Still Alive With Sir Clive!

ZXir QLive Alive!

The Timex/Sinclair NorthAmerican User Groups Newsletter

Volume 10 No. 4 Winter 2001

MEMORY MAP

ADDRESS	Routines
2	Information and Chairmen — Trea$ury Note$
3	*Input/Output — by Abed Kahale*
5	**QUANTA**
6	Timex 3256
7	**OS-64** — David Solly
8	Sinclair E-Mail List
	FILES
9	**QL Hacker's Journal**

ADDRESS	
	FILES
11	**FORTH** — Glen Haydon
13	TS – ZX Spectrum — David Solly
15	EZ80 Micro Computer — Wilf Rigter
20	**Z-88** — Dave Bennett
	SUBROUTINES
21	Unclassified **Ads**

T/SNUG Information

We wish to support the following platforms : ZX-80/81, TS-1000, Spectrum, TS-2068, Z88 and QL. If you have any questions about any of these fine Sinclairs, contact the:

Chairman

Chief Motivator
Donald S. Lambert
738 Gunnar Ln.
Forsyth, IL 62535
(217) 875-8043

Vice-Chairmen

Tape & JLO PD Library
Luke Perry
3708 NE 109th Ave 115
Vancouver WA 98682

Z88 Library

Dave Bennett (HATSUG)
1275 Timber View Dr.
Mechanicsburg, PA 17055-9146
717 732-4374

QL Hacker's Journal

Timothy Swenson
2455 Medallion Dr.
Union City, CA 94587-1914
swensontc@geocities.com

TS-2068

Rod Humphreys (VSUG)
10984 Collins Pl.
Delta, BC V4C 7E6 Canada
604 583-2819

QL PD Library

John Donaldson (CATUG)
835 Foxwood Cir.
Geneva, IL 60134-1631
630 232-6147

AERCO & Z80 Emulator

Keith Watson
41634 Amberly Dr.
Mt. Clemens, MI 48038

--=GATOR=--

Bob Swoger (CATUG)
613 Parkside Cir.
Streamwood, IL 60107-1647
630 837-7957 Work 847 576-8068

ZXir QLive Alive!

Is the newsletter of T/SNUG, the Timex/Sinclair North American User Groups, providing news and software support to the T/S community in a VOLUME of four newsletters per year; beginning with the Spring (March) issue.

T/SNUG's main goal is to preserve and encourage the use of Sinclair computers by providing an open forum for the exchange of knowledge, building and maintaining of software libraries. Providing vendors, repair service and members with free ad space.

It is the user groups and individual subscribers, rather than the vendors, that provide the pecuniary support for this newsletter. Vendors and developers receive this newsletter free of charge, though contribution from vendors and user groups is gratefully accepted.
Please support our vendors and service providers whenever possible.
If you have a problem or you have solved a problem, please share it with the rest of us.
No problem will be considered unimportant.

Editor/Treasurer/Publisher

You can keep T/SNUG alive by an annual contribution of $12 for one VOLUME made payable to Abed Kahale. Send check to:-

ABED KAHALE
232 WEST OAKS TRL
WOODSTOCK GA 30188

NEW

Please do not send mail to the above address before February 10, 2001

I am moving into the Atlanta, GA. area *(How I hate moving!!)*

Back copies are available for $1.00 each postpaid.

Article Contributions

Send in your articles by disk, hardcopy mail, or e-mail and your inputs to:—

Abed Kahale
E-mail: AKahale@compuserve.com

Q-Box BBS 810 254-9878
Utica, Michigan

SOL BBS 520 882-0388
Tucson, Arizona

WEBPAGES

http://users.aol.com/clubbbs/tsnug/
http://www.outlawnet.com/~jboatno4
http://www.unixville.com/2068
ql-users@nvg.ntnu.no

TreaSury NoteS

As of January 13, 2001, we have a balance of $712

Input/Output

by Abed Kahale

Hi Abed,

I do have one request and that is someone to explain how to move files from the **Z88 to the PC** in step by step procedure. I have Windows 98 and I use WORD 97 for my word processor. I have tried it a couple of times but no success since I can't find the file on the PC afterwards. I do hope that I can read the files in WORD 97. I suspect that can be a problem but am not sure. I will have to try again since I have learned more since last time. The manual for PC-LINK (the Z88 to PC software and connecting cable) states that Lotus and Wordstar are used. The manual was written in 1988 so much has happened since then. I need a way to use the Z88 for note taking since my handwriting is bad. Even I sometimes can't read it even if I print.

Sinclairly yours,

Don Lambert
dslambert@email.msn.com

Dear ZX-TEAM friends, please excuse my German mail to you. (Now I have created an English and a German distribution list, so I hope you will never more receive German mails ;-))

I would like to ask you to support ZX-TEAM-MAGAZIN with articles, small unclassified ads, questions and what ever you think, we should print about you and your doings with ZX-81.

With issue 4/2000 we will start out 10th year!

Some dates of interest: 16 Sept. 2000 Sinclair meeting in Utrecht/Netherlands, I hope to come Spring 2001 5th ZX-TEAM-meeting.

Peter Liebert-Adelt
P.Liebert@t-online.de

I am still looking for Window Print 2000 on tape (the DOCs as well if possible but no problem if not) of someone knows of a snapshot posted someplace I could probably use that since I use an emulator also.

I am also looking for a **LarKen** or **Oliger** Disk Interface (combined I hear is nice but do they exist?)

I have a drive already, just need the controller, the cable(s) and the OS cartridge ... I wonder if the Oliger can run an OS-64 at the same time? Thanks!

Robert L Gilbert
dstar33@juno.com

Abed;

Sorry I haven't contacted you before now. I have not had much time to work on the interface over the last few months.

About two months ago I decided that my interface would be most useful if used to support **CP/M** Plus on the **2068**. In the last two weeks I have been working on the BIOS routines for a 64 column screen and keyboard. I have nearly completed the screen BIOS routines and will next will be completing the keyboard and IDE routines in that order. After I get CP/M Plus up and operating I will try porting over the Z-System as it has more advanced features. I have been conversing with Gaby Chaudry in Munich about the Z-System and an enhancement to the interface.

Jeff Burrell
JBurrell@endocardial.com

.....I am always willing to answer any questions about how I use the old **TS-1000** with my trains. You could mention my web site since it has the page devoted to the Timex and Byte-Back I/O board if you would like, but it is practically the same info you have already included in past issues.

Joe Rampolla
http://yourpage.blazenet.net/jprampolla

Dear friends,

In the past months there have been some shortages in new QL hardware.

Additionally there has been a **lot** of discussion for the potential future of the QL. Sooo I just have to show you something **without** any other comment. Just tell me what you think :-)

Here's a REAL address that you should look at ...

http://virtuals.atlant.ru/peters/e-index.htm

http://www.algonet.se/~rsm/zx/clones/peters.html#sprinter

It's info about the "Sprinter" computer ... extended ZX-Spectrum architecture!

Al Feng
alfeng@juno.com

There seems to be a successor to the QL called the Q40. Click on the link on the T/SNUG homepage to the QUANTA homepage. ---==GATOR==---

Robert Swoger - Senior Product Designer
Voice 847 576 8068 Email: CENG108@email.mot.com
bobs@comm.mot.com bobswoger@juno.com
clubbbs@aol.com

Hello Abed,

I hope things are well with you. I am writing for two reasons. First is I would like to re-subscribe to the T/SNUG Newsletter. Please tell me what amount and where to send the check.

Secondly, I'd like to announce that the one and only unofficial **Timex Sinclair 2068 Website** is now operational! After a 2 year hiatus, I've finally gotten on the ball and assembled the site with the information I had available. This is material I've been collecting over 2 years, so there is quite a bit of info to be had.

I've got the official history of the TS-2068, peripherals that will work with it, programming howto's, and much more. I hope to eventually grow the site into a highly useful resource for the rest of the TS2068 users to enjoy.

As always, I welcome comments and material to add. The site will be a work in progress, as all websites are. I recently got a digital camera (very nice) and I've photographed most of my personal collection.

Thanks to Johnny Red, of Portugal, and Jack Boatwright, who have supplied lots of the information, images, and other tidbits for the site. Finally, here's the link:

http://www.unixville.com/2068

Hope you will enjoy it. I will be asking the T/SNUG group how to move TS2068 files over the internet- my software collection is quite lacking at this time. I would also like to post the various disk ROMs, if they're available as files (LarKen DOS, Aerco, etc.) so that you could take the schematics from the Website and make your own Aerco drive interface.

One more tidbit: I will be restarting the 2068 email list once again in a few weeks. I still have many of the original email addresses and will try to contact the individuals. Many thanks!

Louis Florit
florit@unixville.com

Hello everyone, if you receive this email, it means you were originally subscribed to the Timex Sinclair 2068 email list that went offline around the end of 1999.

Everything has been rebuilt, and the email list is back online! You can send a message to all the rest of the people subscribed to this list by emailing 2068@unixville.com

I would also like to announce that the unofficial Timex Sinclair 2068 Website is up with lots of images, information and links to many interesting resources. If you have any articles to post, I would sure appreciate it if you dropped me a link.

The Website is at
http://www.unixville.com/2068

Thanks, and hope to see lots of messages on the pipe again.

Louis Florit

Hello....

I am wondering who among you has used the Warjevo Emulator and has any remarks -- good or bad -- about it. I am especially interested in how the program loads software from tape and if there is any sort of microdrive emulation. Also, I would like information on where to obtain the latest version. Thank you all in advance.

David Solly
Zabad@freenet.carleton.ca

I believe the latest version should be available from the Website

http://www.void.jump.org/warajevo/

I haven't used it myself, because of the fact that it is dos only. (No gui/windows). According to miscellaneous reports I've read, its supposed to be one for the most accurate and quickest emulators available.

Louis Florit

Hi,

I have used the **Warjevo Emulator** a lot and like it very much. It does support the microdrives in the Spectrum mode, and with a special driver it can support the microdrives in Timex 2068 mode. The reason I like it so much is, it can support OS-64 and the microdrives at the same time. This driver is not yet available on the Warjevo Emulation site.

There was another improvement of the Timex microdrive plug-in intended to be uploaded to the Warjevo Emulation site, but it was never done. I don't know why. Zeljko Jurik may have lost interest in the project. Anyway, there are two attached ASCII text files that give some explanation of the use of the Warjevo Emulation in Timex 2068 and OS-64 modes. One, titled SERIES1.TXT, is a copy of the text file already available at the Warjevo Emulator site. The other, titled SER1.TXT, was to be uploaded by Zeljko sometime in mid 1999 along with the enhanced code for microdrive compatibility with Timex 2068 mode and OS-64. I hope attachments are acceptable for this address. Regards,

Keith Watson
2068@unixville.com

Is there an active users group for Timex/Sinclair computers any longer?

I have questions concerning using Tasword II with the LarKen disk system and printing with the Aerco parallel interface to the Legend 808 printer in particular...linefeeds, window width. I have misplaced the documentation that came with the Larken disk system. Isn't there a screen copy command something like PRINT USR 100: LPRINT CHR$ 1 to get a screen copy? That is only if you are in BASIC this command cannot be typed from within Tasword. MSCRPT seems to print fine. Thanks,

Neil
pxx1@netzero.net

Here is what I use to print program listings to a Panasonic printer. For screen COPY, LogiCall by Robert Swoger is your best bet.

```
10 RANDOMIZE USER 100: OPEN #3 "LP"
20 RANDOMIZE USER 100: POKE 16090,132: REM sets left column
30 RANDOMIZE USER 100: POKE 16092,0: REM for right column
40 PAUSE 10: RANDOMIZE USER 100: POKE 16094,1: LPRINT: REM margin for right column
50 OUT 127,30: OUT 127,27: OUT 127,20
60 OUT 127,27: OUT 127,56
70 LLIST
```

Hi Abed,

I redid my opening page on my site with a new animated photo. If you have the time, take a look-see.

Joe Rampolla
http://yourpage.blazenet.net/jprampolla
jprampolla@blazenet.net

I had a look at the site you marked. I believe that TC3256 is the "Portuguese" Timex. I understand that it is almost an exact copy of the T/S 2068 except that all the messages are in Portuguese, the video output is for monitor or uhf and the bus is a ZX Spectrum bus rather that the T/S bus that was used in the US. Hope this helps

David Solly

The TC2068 doesn't have the messages in Portuguese. The TC2068 have the TS2068 ROMs and 48K RAM. The TC3256 was a new computer, new ROMs, new hardware. It had 256KB RAM, network, ZX Spectrum compatible mode, etc... To know why Timex of Portugal pulled the plug, read the Timex of Portugal part in the Timex History page.

Johnny Red, Portugal

Timex Computer World
http://homepage.esoterica.pt/~joinred or
http://www.colprinter.com/tmx

Keith Watson:

I have used the Warjevo Emulator a lot and like it very much. It does support the microdrives in the Spectrum mode, and with a special driver it can support the microdrives in Timex 2068 mode. The reason I like it so much is, it can support OS-64 and the microdrives at the same time.

I've used it quite a bit too. Its strongest feature, what makes it special IMO, is the built in monitor/disassembler. I used it a few times to try out some TS2068 snapshots (MSCRIPT, techdraw, zeus assembler, etc.) and they worked great. The only problem is, you can't load the snapshots from within Warajevo - for some reason the emulator changes to Spectrum mode when it loads snapshots. I had to start up the emulator with "TS2068 /rMSCRIPT.sna" or something similar.

Alvin Albrecht

Alvin....

The only version of "C" for the ZX Spectrum I have seen and used is the HiSoft version. The only useful thing I have see about it is that it comes packaged with an extensive graphics package which might be great for game developers, otherwise, it is pretty useless. It has only integer addition and subtraction and no built-in function for multiplication or division. To multiply or divide you have to poke two system variables with values and call the ZX Spectrum ROM routine. It also uses the hated and unreliable 128-byte block method for loading itself and for source or object code you want to save or load. I believe there is one other version of "C" available for the ZX Spectrum but I have never seen it.

For the record, I have tried Abersoft and Hawg Wild FORTH. It is still integer math only, however, you can do 16 byte math by using specialized functions. For Abersoft FORTH the storage device is strictly tape with no ability to change the file name. It saves everything as "DISC" and that is the only choice you have unless you want to write a new function. Hawg Wild FORTH gets most of its functionality by hooking into the T/S 2068 ROM routines which in some cases makes it slower that calling the equivalent routine in Sinclair BASIC. There is also Sinclair FORTH which I have never been able to load into my ZX Spectrum emulating T/S 2068 but I have been able to make it run under Gerton Lunter's Z80 ZX Spectrum emulator.

David Solly

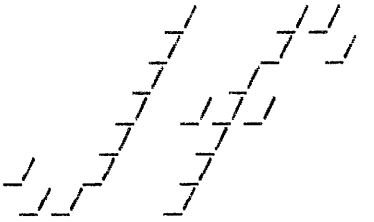
Hi David,

The one I'm thinking about is available here: http://z88dk.sourceforge.net/features.html
It's a cross compiler, meaning you run it on your PC but it generates a code block for a z80 machine. It does support floating point using its own routines (or stubs, which I presume you can fill in with calls to the 2068 ROM fp calculator, if desired) and it has a decent subset of ANSI-C (or it seems to me..) including support for far-pointers: a method for dealing with bankswitched memory.

It looks nifty but I haven't had time to really check it out. Dominic started a library of routines for the Spectrum (which we could modify for the TS2068) but his interests seem to lie with the Z88 nowadays so a person interested in this C compiler would do us a favor by working on a TS2068 library of routines :-).

Alvin Albrecht

I did this with a TS1000 a long time ago. You pulled it off by switching from FAST to SLOW mode and as I recall, the PLOT command (or anything that wrote to the screen) would do it. It's been years since I did this but with a little experimentation you should be able to duplicate it.



jfarr@icubed.com
http://www.icubed.com/~jfarr

Quanta

Hello

We are pleased to announce a major upgrade to the original QL Users Email Database now found on

www.quanta.uni.cc

New features include much cleaner and simplified access, and a communications facility. The database at the time of writing, has some 388 entries, with numbers steadily increasing.

Access to database functions is now achieved directly from Quanta's homepage (which incidentally has also been completely re-hashed!). For instance, if you have forgotten your password or simply do not have one, all you do is enter your email address on Quanta's homepage, select the forgot/none option, and press Go!

The communications facility has been introduced to keep up with inter-computer voice and chat communications. Two options are available Both use a pop-up window thus allowing you to continue surfing the net. The Comms option displays details of other users logged on, giving details such as their IP address and software package/ID employed for direct voice communications. The QL Users Chat option additionally displays text input and output windows.

Since it would be ideal to know when other users might be logged on, you can place a diary entry in your database record and this will be displayed on Quanta's homepage.

Quanta Links Page.

This has been removed due to misuse by advertising organizations on the net. A new links list is now generated however from your database URL and site name entries. If you placed URL details on the old links page, these will no longer be visible, so update your database entry.

More on Database Entries

If you have not yet updated your entry since the databases relocation to www.quanta.uni.cc, please do so. if you do not have a password simply go to www.quanta.uni.cc and get one. Make sure you enter the email address this letter

has been sent to! Some optional database entries include for instance, Quanta and QL Today membership. Completing these entries are quite important because not only will these details be used when sending information to different interest groups, some future facilities will only be available to certain interest groups. Regards.

Quanta Support

This email has been generated using the bulk email facility at www.quanta.uni.cc. In the event of a complaint or enquiry contact

quanta_support@uk2.net
wwwrun@www.jakinternet.co.uk

Timex Computer 3256

Timex Computer 3256 had really existed! Not a prototype, but a fully functional and finished machine! It was totally developed by Timex Computer of Portugal. TC3256 have 256K base memory and a keyboard like the actual PC's. It was based in Z80A CPU and it have five modes of operation.

1-BASIC This is the Spectrum mode.

2-TIMEWORD

TIMEWORD is a word processor in ROM. It uses 80 columns screen, have all the Portuguese charters. It can use the RAMdisk, FDD or tape for storage. Have TABS, copy and move blocks of text.

3-Terminal CP/M This mode was to replace the TT3000. It sets the TC3256, to use the FDD3000 in CP/M mode and it works in 80 columns.

4-Disk Start

This gives to the programmer 64K of RAM to development of software. The biggest advantage is that the 64K RAM starts at 00H.

5-CATRIDGE

Specially designed for the software houses to protect the software, using cartridges. TC3256 have a Network, the TENET (Timex Educational NETwork)

- 25 stations
- Maximum length: 100 meters
- Sharing Printer and FDD System
- Messaging system
- Can be used in Timex Extended BASIC
- Station low cost fully transparent to users software
- Resists to failure or loss of stations

Timex Extended BASIC has added more commands:
LOAD!,SAVE!, CAT!, MERGE!, ERASE! and CLEAR! to control the RAMdrive.
FORMAT!, LPRINT, LLIST to control the RS232
BEEP! x,y to control AY-3-8192 sound chip
SCREEN$, DRAW!, PLOT!, CIRCLE! to control the high resolution screen mode. it allows to work with PAPER and INK.
TC3256 doesn't need anymore the FDD/FDD3000 interface. The FDD3000 is just connected like the TT3000.

TC3256 Specifications
Processor: Z80A
RAM: 256K (208K RAMdrive/48K base memory)
ROM: 64K
16K Sinclair BASIC
16K TIMEWORD word processor
16K Timex Extended Basic (Tenet, disk, RAMdrive)
16K CP/M Terminal emulator
Screen: 32x24 / 64x24
Resolution: 256x192 / 512x192
Sound: BEEP and AY-3-8192 (can output to TV)
Joystick: 1 Kempson
Cartridge Port: 1
Disk Drive: TOS / CP/M
TV Output
Monitors: Video composite and RGB (colour/ monochrome)
Colours: 8+BRIGHT=15
Keyboard: 69 keys, professional keyboard with numeric keyboard and function keys (Caps Lock, Extended, cursors, edit delete, break)
RS-232: 1 (300 to 9600 baud)
Mic and Ear: Can use Tape

Timex of Portugal stop selling, repairing and manufacturing all the Timex Computers range when the TC3256 was about to be launched in the market. The destiny of TC3256 is unknown, but I think I'll get this info, later!

A Portuguese guy have send me an email telling that he have a working TC3256. He said that he would get pictures of it to me.

Johnny Red
Portugal Timex Computer World
http://homepage.esoterica.pt/~johnred or
www.colprinter.com/tmx

Here is a translation to the text on the TC 3256 that the link below refers to. TC 3256 (Computer, Portugal)

Z80A CPU, 256 Kb RAM, 64 Kb ROM (16 Kb Sinclair BASIC, 16 Kb Timeword, 16 Kb Timex Extended BASIC (TeNet, Disk and RAM-Drive), 16 Kb CP/M Terminal Emulator), 32 x 24 and 64 x 24 characters respectively 256 x 192 and 512 x 192 pixels (8 colours + BRIGHT), five operating modes:

1. BASIC - Spectrum mode,
2. Timeword - Text processor with 80 characters per line, uses FDD or cassette for mass storage,
3. Terminal CP/M - This should replace the TT3000; uses the FDD3000 in CP/M mode,
4. DiskStart - Gives the developer 64 Kb of free storage from address 0000H, and,
5. Cartridge - Software is loaded from cartridge.

Louis Florit wrote:

Anyone ever hear of a TC3256? This site makes mentions of it (in German) [and gives a nice Timex Sinclair/computer timeline]

http://museum.ruhr.de/docs/timex3.htm

Louis Florit

Dear Abed,

Please find with this letter a copy of TimexSinclair

ZX Spectrum ROM Routine Look-Up Table

for publication in the next edition of ZXir QLive Alive! I have referred to this table a few times myself and I hope readers of ZQA! will find it useful. However, cannot take credit for having created it. It is something that was photocopied from somewhere else and given to me. I have merely made a new copy by scanning the original and doing some "fly spot" removal. (Someday, when I have the time and ambition, I might enter all this data into a database and create a two-way table.) Your friend,

David Solly

(See the next pages)

Hello....

I would just like to remind people that the following LarKen manuals are available on the World Wide Web. Larry Kenny. LarKen disk editor, edited and annotated by David Solly. HTML conversion by W. McBrine. Ottawa. Bibliotheca Sagittarii, ©1999. URL:

http://www.clark.net/~wmcbrine/larkdisk.html

LarKen disk operating system, Version L3F: operating manual / edited and annotated by David Solly. HTML conversion by W. McBrine. Ottawa: Bibliotheca

Sagittarii, ©1998. URL:
http://www.clark.net/~wmcbrine/larkendos.html.
LarKen sequential/random access file utility, edited and annotated by David Solly. HTML conversion by W. McBrine. Ottawa: Bibliotheca Sagittarii, ©2000. URL:
http://www.clark.net/~wmcbrine/larkenseq.html.
In addition, I am currently working on two other LarKen manuals. LDOS for the TS-1000 Manual & LDOS for the TS-2068. For the TS-2068, LDOS was the forerunner of LKDOS. I don't know if William McBrine will make them available on his web site because I have yet to hear from him. I will post a notice to the news group as they become available.

David Solly
Zabad@ncf.ca

Gentlemen,

I received a question from a man from US News & World Report asking if there is still a Timex User group out there. He's considering an article about us diehards. In my response I mentioned the ZXir QLive Alive! newsletter as well as the existence of NESQLUG and the Cleveland group to him. I hope you won't mind that I gave him your email addresses. You may be getting email from Al Weiner shortly so this may give you some time to think about it.

John Donaldson
jldnldsn@aol.com

This site (www outlawnet com/~jboatno4/welcome.htm) is the only place on the web I got a hit for Gladstone Electronics. Do you happen to have a current phone or address? I am trying to find a transformer for some audio modules I purchased from them in the mid 80's. Thanks, any help would be appreciated.

Jeffry Sczepanski
jsczepan@ford.com

Sorry, I don't have any info on Gladstone.
But you can try Radio Shack of course.
All Electronics - www.allelectronics.com
MCM Electronics - www.mcmelectronics.com
Mouser Electronics - www.mouser.com
James – www.jameco.com Newark.....

Does anyone have a **TS-2020** tape player that they'd care to part with for relatively cheap (25$)? You can reach me at florit@hudat.com

Louis Florit
florit@unixville com
LOUIS FLORIT
160 PASITO TER 718
SUNNYVALE CA 94086

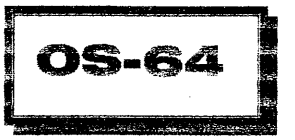
Hello, does anyone have any experience using the terminal program made for the OS 64 cartridge? And if so, can you use it independent of the cartridge or is the cartridge needed? I would like to get a different terminal program for my TS2068, the MTERM software program is very limited thanks,

Luke Perry
doidy34@yahoo.com

Thanks for the great info David. I would love to get the OS 64 cartridge and ZTerm terminal program for my TS2068. If anyone on this mailing list has this for sale/trade please let me know. Also, is the Oliger disk system compatible with OS 64? thanks, **Luke Perry**

Hello Perry....

The OS 64 cartridge is more properly called the Zebra OS 64 cartridge. It is designed to fit into the cartridge dock of the Timex/Sinclair 2068 Colour Computer. However, because my T/S 2068 shares the same dock with the LarKen DOS chip, I had built an arrangement that piggy-backs the Zebra OS 64 chip over the LarKen DOS chip. A small toggle switch is used to switch the Zebra OS 64 system in or out. The Zebra OS 64 chip is also completely compatible with LarKen DOS.



The terminal program written for the Zebra OS 64 is called ZTerm. ZTerm is specifically designed to run under Zebra OS 64. It is also designed in such a way that it can only be run using the Westridge 300 baud modem. All attempts that I know of to patch it to run on a standard modem failed. Despite this, it was an excellent telecommunications program for the time. Because of the Zebra OS 64 chip, ZTerm can display 64 characters per line in full sized character, i.e. 8 by 8 pixel characters, which is a great deal less stressful on the eyes than the compressed character sets used by such applications as Tasword II. You can also install a customized character set, if you so wished. I used the Amstrad character set myself for terminal work. Other features of ZTerm are the ability to create and store address books, up and download programs using x-modem protocol, capture text to a text buffer and program function keys.

Shortly after the appearance of ZTerm, a program from Britain called Specterm made its appearance. Specterm has most of the features of ZTerm, however, it requires your computer to be fitted with a ZX Spectrum emulator chip. Specterm is not as stable as ZTerm and tends to hang up easily. Specterm also uses a compressed character set to obtain 64 characters per line. This character set is very stressful on the eyes -- even with a good monitor -- after only a very short time. Where Specterm outshines ZTerm, however, is that it comes with several overlay packages which allows it to be used with several mass storage devices and also, with the aid of an RS232 board, with standard Hays compatible modems in addition to the Westridge modem. Specterm also used x-modem protocol for file transfers but, in addition, it sends header information with the file. This is both a blessing and a curse. A blessing in that you do not have to worry yourself over whether the program was ! BASIC, machine code, a screen save or a data file. The header information insured that the program will be saved in the correct format to whatever storage device you are using. The curse is that both parties must be running Specterm for this to work otherwise you get garbage instead of a program.

I hope this information is helpful or at least provides an interesting history lesson.

David Solly Zabad@ncf.ca

If I remember correctly, Zebra Systems is still in business and the Zebra OS 64 chip and ZTerm program is still available from them. Zebra is one of the very few business left that still supports Timex/Sinclair. Cheers,

David Solly

Sinclair E-Mail List

Anderson, Paul	pandersn@peakpeak.com
Anson, Gerald	jerrya@aztec.asu.edu
Barker Robin	robin@di-ren.demon.co.uk
Bennett, Dave	daveb357@juno.com
Bennett, Dave	daveb3572001@yahoo.com
Boatwright, Jack	jboatno4@outlawnet.com
Boehm, Al	boehm@ziplink.net
Boehm, Bill	boehm@plh.af.mil
Burrell, Jeff	jburrell@endocardial.com
Burt, Richard	ajb@intranet.ca
C. A. T. S.	mf0002@epfl2.epflbalto.org
Cable, Bill	cable@cyberportal.net
Carpio, Juan	juanchuscar@yahoo.com
Castro Antonio	castrox@portoweb.com.br
Catotti, Christopher	kd4ace@compuserve.com
Chambers, George	gfchamb@pathcom.com
Collins, Bill	bcollins@home.ifx.net
Cottrell, Les	jacottre@gte.net
Cruz-Figueroa, Jaime	cruzfiguer@aol.com
Dansby, Andrew	adansby@atlantic.net
Davis, Frank	fdavis@iquest.net
Delhez, Carlo	carlo@spase.nl
Donaldson, John	ildnldsn@aol.com
Dorinson, Mark	74200.257@compuserve.com
Dunbar, Douglas	dldunbar@prodigy.net
Dunnet, Ron	ron@qubbesoft.freeserve.co.uk
DuPuy, James	dupuy@pipeline.com
England, William	wengland@iname.com
Feng, Al	alfeng@juno.com
Fink, Mike	domino.cubes@excelsior.net
Fink, Mike	domino.cubes@pointblank.com
Firshman, Tony	tony@firshman.demon.co.uk
Florit, Louis	florit@unixville.com
Franke, John	j.m.franke@larc.nasa.gov
Ganger, Gary	gangerg@dma.org
Gilbert, Robert	weena@netzero.net
Gillespie, Doug	aa431@cleveland.freenet.edu
Girnius, William	girnius_w@bls.gov
Goodwin, Glen	glenatacme@aol.com
Gowen, Rod	aw723@osfn.org
Haberly, Duncan	duncan@military.com
Haberly, Duncan	duncan@military.com
Harbit, Ken	krh03@cvip.fresno.com
Henderlight, Mike	mikehend@microsoft.com
Henn, Fred	oranur@juno.com
Herre, Cy	cyherre@aol.com
Holmgren, Paul	paulholm@indy.net
Horton, Will	willhort@aol.com
Hoshor, Dave	dnhoshor@raex.com
Humphreys, Rod	rodh@lightspeed.bc.ca
Hunkins, James	jdhunki@ibm.net
Impellizerri, John	jimpellizerri@compuserve.com
Jaap, Matthias	matthias_Jaap@hhs.hh.schule.de
Jonas, Mike	mjonas@bbn.com
Jones, Dilwyn	dilwyn.jones@dj.softnet.co.uk
Jones, Terry	tjones@iname.com
Kaczor, Jon	jazkaczor@aol.com
Kahale, Abed	akahale@compuserve.com
Kealy, Harriet Joan	hjkealy@admin.hilconet.com
Kenny, Larry	larken@storm.ca
Kingsley, Ed	edk4@aol.com
Knyszek, Theodore	thirteenth@worldnet.att.net
Kondrac, Mike	mkandrac@aol.com
König, Urs	urs.koenig@agrodata.ch
KurtK7	kurtk7@aol.com
Kwitkowski, Phillip	kwit80@aol.com
Lambert, Donald	dslambert@emailmsn.com
Lancaster, Garry	dharkhig@delphi.com
Lanciault, Francois	francois.lanciault@energies.alstom.ca
Lassov, David	emanon@azstarnet.com
LaVerne, Melvin	mlaverne@usit.net
Lebowitz, Dave	dkl@dpliv.com
Lessenberry, Gary	gl743@aol.com
Lewis, Jim	jlewis2@neo.lrun.com
Liebert-Adelt, Peter	Peter@zx81.de
Liebert-Adelt, Peter	p.liebert@t-online.de
Malloy, Bob	74776.2342@compuserve.com
Matthias, Jaap	mjaap@atari-computer.de
McBrine, William	wmcbrine@clark.net
McKelvey, William	mckelveyw@delphi.com
Merz, Jochen	jmerz@t-online.de
Mikolajczyk, Dean	deanm97493@aol.com
Miller, Seymour	seymil@delphi.com
Mills, Frank	effem417@yahoo.com
Muth, Bob	bobkeeper1@aol.com
Norton, Gary	gnorton@world.std.com
Parrish, Gil	gil.parrish@abanet.org
Pashtoon, Nazir	nazir.pashtoon@ingram.micro.com
Payne, Josh	joshpayne@bigfoot.com
Pazmino, John	john.pazmino@moondog.com
Perry, Luke	Luke_Perry@gstworld.net
Perry, Luke	doidy34@yahoo.com
Perry, Russ Jr	slapdash@enteract.com
Radloff, Toby	tobyradloff@webtv.net
Rampolla, Joe	jprampolla@blazenet.net
Rigter, Wilf	rigter@cafe.net
Rish, John	74601.1535@compuserve.com
Shepard, Jay	jshepard@wccta.net
Simon, Thomas	73177.333@compuserve.com
Skapinski, Thomas	tskapins@juno.com
Solly, David	zabad@freenet.carleton.ca
Stegman, Dan	danesteg@juno.com
Swenson, Tim	swensontc@geocities.com
Swenson, Tim	swensont@sirclive.csd.sgi.com
Swentko, Wally	wswentko@maroon.tc.umn.edu
Swoger, Robert	ceng108@email.mot.com
Taylor, Jeff	jetaylor@mdrobotics.ca
TEJ Computer	tej@jps.net
Thoresen, Jeff	74200.257@compuserve.com
Waldman, Stephen	brogine@hotmail.com
Walterman, Don	walterm@ix.netcom.com
Watson, Keith	keith_watson@juno.com
Wood, Roy	qbranch@qbranch.demon.co.uk
Zimmerman, George	gzimmer928@aol.com

QL Hacker's Journal

Supporting All QL Programmers #33 October 2000

The QL Hacker's Journal (QHJ) is published by Tim Swenson as a service to the QL Community. The QHJ is freely distributable. Past issues are available on disk, via e-mail, or via the Anon-FTP server, garbo.uwasa.fi. The QHJ is always on the look out for article submissions.

QL Hacker's Journal
c/o Tim Swenson
2455 Medallion Dr.
Union City, CA 94587
swensont@lanset.com
http://www.geocities.com/SiliconValley/Pines/5865/

Editor's Forum

I don't know if I'm old enough to claim that old age is to blame for why I'm not able to get as many QHJ issues out in a year as I would like. Maybe I can blame my other distractions, such as family, house, work, and my other hobbies. I've also been a little distracted with the Q40 and Q40/Linux.

So, here I sit not doing any coding on the Q40, but just thinking about coding. This issue is mostly on things to help code, but no actual code. As I putter around the house, fixing this and working on that, I keep thinking to myself; "I'll get a chance to work on that program tomorrow." Then tomorrow comes and there is more distractions. Ahh, Well.....

Turbo and TurboPE

In the May/June 2000 issue of QL Today, George Gwilt mentions that he has updated Turbo to work under SMSQ/E. He also mentions creating TurboPE, a Pointer Environment interface for Turbo. He does not go into any details nor give any code examples, so I can't say how easy TurboPE will be to use over other PE tools (QPTR, EasyPTR, and Qmenu). He does not even mention if these existing tools will work with Turbo.

Given that Turbo will be freeware, I'm making a fair guess that TurboPE will also be freeware. As useful as QPTR, EasyPTR, and esp. Qmenu are, they are all commercial programs. It is possible to write freeware applications with these tools, but distributing a fully working program is limited. Qmenu may not be distributed with a freeware application, EasyPTR routines may be compiled into an application, but permission must be granted for each application, and QPTR does not mention if it can be compiled into an application. Granted the PE can not be freely distributed so any freeware PE programs must assume the user already has the PE, only Qmenu is distributed with other programs.

So having a freeware SuperBasic compiler with a freeware toolkit for accessing the PE will give the QL programmer a little more freedom in what he can write and distribute. I look forward to the release of both Turbo and TurboPE.

Recent Commercial to Freeware Released

Recently a number of commercial programming tools have been released as freeware. They are ProWesS, DJToolKit, MasterBasic, and the TurboToolkit.

Of the four packages, ProWesS is the biggest and most surprising to be released as freeware. It is a major piece of work, fairly complicated, and gives a lot a capability to the user (and programmer).

Let me take a minute and give a quick description of ProWesS. I'll start with something most of you would already know about, the Pointer Environment. The PE is composed of two parts, the Pointer Interface (ptr_gen) and the Window Manager (wman). The PI takes care of the mouse and the WM takes care of the screen displays. The look of the PE is based upon the Window Manager (wman). ProWesS is another Window Manager for QDOS. It is used to create a whole different look for applications and provides some features (such as scalable fonts) not found in the PE Window Manager.

Another way of looking at this is to view Windows 3.11 and Windows 95 as two different Window Managers. A program written for Win 3.11 will run under Win 95, but it will have the look of a Win 95 program. For users of Q40/Linux, they are finding out that Linux has quite a number of different Window Managers available.

Unlike most Window Managers, PE and ProWesS programs can be run at the same time and be displayed on the same screen.

For the programmer, what ProWesS gives is another way to create pointer driven programs. If programming for the PE is a little beyond a programmer, they might find programming for ProWesS a little easier. Plus, the various additional features that ProWesS has over the PE gives the programmer additional capabilities for the application.

The DJToolKit (DJTK) is a collection of SuperBasic extensions written by Norman Dunbar. I bought DJTK a few years back and have found it useful in my programming. It contains 44 new keywords, which are broken down into 4 main areas: File handling, Font handing, Screen handling, and Heap handling. The distribution comes with the complete documentation, giving enough information to get started. I'd recommend that all SuperBasic programmers give this toolkit a look. You might just find THE extension that makes your programming easier.

TurboToolKit (TTK) has been updated to fix a number of bugs and to have it work with newer versions of QDOS and SMSQ/E. Like the DJTK, TTK is a collection of extension for SuperBasic. I have never used Turbo or the TTK so I don't know any specifics about the extensions. Some of the extensions are document in the text files that come with the distribution,

but they look to cover those parts of TTK that have been updated. Like DJTK, now that TTK is freeware, it's worth a look to see what it has. Maybe better documentation will be released with Turbo is released.

MasterBasic by Davide Santachiara (and released by Ergon Development) is a SuperBasic tool for assisting in the creation and debugging of SuperBasic programs. This program has recently been released as freeware and is available on the Ergon Development web page .....

I must admit that I bought this program a few years ago, but have not really gotten around to trying it out. MasterBasic is a collection of routines and programs with a graphical front end. It is designed to work on a program in memory (i.e. a LISTable program). It can do things like searching for a string in the program, quickly find and edit Procedures and Functions, quickly finding a variable, and so on. It also comes with some accessory utilities like a calculator, notepad, and job management tool.

I guess the reason I have never tried MasterBasic is that I prefer to write SuperBasic in an editor (like MicroEmacs, ED, etc) and then LOAD and test it. With an editor I can quickly jump around putting in bits of code as I think of them. I can easily put more verbose comments in the code. Add white space for easy reading. I don't plan my code to be syntactically correct when I write it. My first draft is very similar to a rough pencil sketch. Once I've done the mental dump to the editor, I can then flesh out the code and make sure it's good SuperBasic.

LibXmenu

LibXmenu is a C68 library, written by Jerome Grimbert, to help write menu-based programs. When I first heard of it I was thinking that it was a library that converted X-windows routines to PE routines, but this is not what it does.

Jerome described Xmenu as a collection of C routines that he wrote to assist him in writing PE programs. He found himself doing the same routines, so he just created some fairly generic routines that could easily be used from one program to the next. Looking at the routines available, it looks a lot like Qmenu for C68. Jerome did tell me that this was not his intention.

A quick rundown of the routines are:

Item_Select() - Choose from a list of items.
Item_Select_Array() - Same as above, but using an array.
Message_Report() - Display some text.
String_Edit() - Get a string from the user (with edit).
XDialog() - Display sprite and text.
Menu_Button_Text() - Text for the Button.
Menu_Button_Logo() - Use a sprite in a Button.
List_Select() - Choose 1 or more from a list.
List_Select_Array() - Same as above, but using an array.
Check_Size() - Find size of window.
Get_CharSize - Find size of characters.
DefaultColourSet() - Change color set to default.
DisplayXSize() - Maximum X value of current display.
DisplayYSize() - Maximum Y value of current display.
SetWindowColour() - Set color of the window.
SetInfoColour() - Change color of the info window.

These routines can be used in both Mode 4 and Mode 8. On Jeromes' web page he has a page describing libXmenu with example screen shoots showing the results of some of the commands. I'm not much of a C68 programmer (heck, I hardly have enough time to program in SuperBasic), but if I was to start, I would start playing around with libXmenu to see how easy it is to write a menu-based program.

Recent Upgrades To Microemacs

MicroEmacs is becoming my favorite editor, esp. now that I have a Q40. As MicroEmacs becomes more powerful, it needs some more horsepower to run quickly. On the Q40, speed is not an issue at all.

There are some new features to MicroEmacs that impact the programmer. They are:

- Support for Client-Server Manager (CSM)
- Syntax Highlighting
- Support for CTAGS

CSM is an easy way for one program to control another. The Server program handles requests from the clients, and passes back to the client an acknowledgement of the request or some data. When I first encountered CSM I was not too sure how it could be used. It was created to provide scripting for QEM. The scripts are written in SuperBasic, sending commands to QEM via CSM.

Now MicroEmacs has CSM support, allowing SuperBasic programs to send MicroEmacs commands to the editor. This opens up a number of possibilities for having other program interact with MicroEmacs.

How I am thinking about using the MicroEmacs-CSM link is in Structured SuperBasic (SSB). SSB is just a simple filter that converts SSB code to SuperBasic. If SSB encounters a syntax error, it just reports it and exits. With a link to MicroEmacs, SSB could connect to MicroEmacs via CSM, move the cursor to the exact line with the error, then SuperBasic can exit. The user would switch to MicroEmacs and be at the problem line, ready to edit it. This would move SSB and MicroEmacs closer to an Integrated Development Environment.

Here is an Structured SuperBasic example of how easy something like this would be.

```
## Make sure MicroEmacs is running as a server
IF FINDSERVER("emacs") THEN
  ## Connect to MicroEmacs
  CLIENT "emacs"
  ## Add line count to goto-line command
  command$ = "goto-line "&var
  ## Send a request to MicroEmacs
  REQUEST "emacs",command$,return$
  IF return$ = "KO" THEN
    ## There was an error
  END IF
END IF
```

Syntax Highlighting is a way for MicroEmacs to show the syntactical elements of a program by showing the different words in different colors and italics. Syntax Highlighting supports C (_c and _h files), SuperBasic (_bas, _sbas, _ssb files), and assembler (_asm, _s, _cmd, and _rc files).

I've only used Syntax Highlighting with Structured SuperBasic and it does make the code a little clearer.
Comment lines (##) are shown in white and italics. There is a slight problem here as SSB defines comment lines as any line starting with ##, excluding any spaces in front of the ## (this allows for indented comments). MicroEmacs only supports lines that begin with ## in column 1.

SuperBasic control commands (DEFine, FOR, TO, NEXT, RETURN, IF, THEN, ELSE, etc.) are shown in white. The SuperBasic short cuts (such as DEF for DEFine) are not supported. The case of the words are ignored. DEFine, DEFINE, and define are all understood to be the same. The rest of the code is shown in the normal green.

The color used to highlight the different elements can be changes by using 4 different MicroEmacs variables. So, if you prefer comments to be in red, you can do that.

Ctags come from Unix and were originally created for the Unix editor, vi. A tag file is created by using the program ctag. The editor then reads this file to understand the tags.
Ctags is designed to be used on a collection of source files in a directory. The tag file is sort of like an index file showing what procedures are defined and used in what source files. Ctags is not very useful with source code residing only one file.

MicroEmacs understands how to read a tag file and can be used to navigate through the source code files, by moving to a procedure and executing the 'tag-word' command. Moving forward is accomplished with the 're-tag-word' command and moving back is done with the 'back-from-tag-word' command.
Ctags understands C, C++, Eifiel, Java, Fortran, and SuperBasic. This means that MicroEmacs also supports these languages with tag files.

THE FORTH ANYONE?

Glen B. Haydon, M.D.

Epsilon Lyra Corporation
Route 2 Box 429
La Honda, CA 94020-9726

I would like to build my Forth Philosophy this year (1998) around the concept that there is strength in diversity. Over the years, Forth has evolved in many divers ways. Individuals have taken different directions and the vendors have gone their own ways. Trying to force all Forth into a single mold will distract from its potential.

Forth as an Operating System

First off, Forth was designed as an operating system. As such it improved the efficiency of the application. However, it restricted the system on which it could be used. Even with that limitation, Chuck brought up most of his implementations on dedicated systems and directed them to specific applications including any necessary operating system. It is interesting that again there is discussion of making Forth the operating system after having taken a turn as only one program available in other operating systems. Perhaps this idea needs closer examination.

Early Public Forth Implementations

Early on, The Forth Primer from the Kitt Peak National Observatory provided a tutorial to the implementations on a set of minicomputers. But it was not until the fig Forth implementation was made available as the Forth Installation manual that the language began to spread.

Several groups were taken with the power of the language and the Forth Interest Group eventually published source code for the implementation of Forth on some 12 different platforms.

At this time the interest in Forth exploded. The monthly fig Forth meetings were well attended. Regularly 150 to 200 were present at these meetings. There were problems and individuals contributed to solutions. The openness of the source and the solutions was stimulating.

Vendor Forth Implementations

At about this time several vendors developed and marketed their Forth implementations. There were numerous discussions about the need for more implementations of the Forth kernels. Each company had a good one. It was time to forget about implementing Forth and get on with the work of Forth applications.

It is interesting that most of the Forth Vendors could not make a business on sale of their implementations alone. As far as I know they all provided custom services for specific applications. Vendors often tailor their kernels for each application.

The early conflicts in goals among Forth users stimulated a wide variety of directions. The differences were always well discussed at fig meetings and conferences. The discussions were lively and comments were at times a little on the sharp side.

The Forth Heyday

This was the time of the heyday of Forth interest. We had a Byte issue which featured Forth. We had nearly 5000 members of fig. The FORML meetings were attended by 100 or so interested Forth users. For a few years we had an annual Forth Day which ran on to two days and were attended by 1000-2000 interested people.

A Change

This review of history is designed to highlight some of the diverse events which led to a rapid growth of interest in Forth. There was something about the philosophy at the time which was positive and stimulating.
Since then, the size of the Forth Interest Group has gradually decreased. We hear less and less about Forth in the general literature. The Rochester Conferences have ended after 18 years.

It is not clear that these FORML Conferences will continue after 20 years. What happened? The Standards Team entered the picture. First with the 79 Standard, then the 83 Standard, and then over a period of years the ANS Standard was developed. At the same time, fig offered no implementations for the new Standards as they came out. This was left to the Vendors.

I see the problem to be that Forth, in contrast with

other languages, was originated as a language closely related to the application and the system being used. Most programmers are not familiar with closely bound hardware and software.

A second problem was that most users wanted to use their computers for other programs too. These systems came with an operating system to manage the other programs.

Forth became just another program. True, by renaming the Forth program COMMAND.COM, the Forth program would become the operating system. However, it did not allow ready access to other programs on the same system.

Then there was demand for Forth Libraries and other tools to which programmers had become accustomed. Programmers new to Forth thought it had to look and work like other languages.

Hardware Evolution

Over the years, the computer systems have become more and more complex. Memory managers have been added under the control of the existing operating system. The disk managers were needed to allocate space on the floppy and then the hard disks in ever increasing size and used for many different programs.

Forth, originating as an operating system closely linking the hardware to the program for optimal efficiency, lost out for many of the major systems on the market.

Windows Operating Systems

Next came the windows paradigm starting with the Star system at Xerox Palo Alto Park. Over years, programmers became accustomed to a windows presentation. The operating system exploded in size and complexity.

Forth became a smaller and smaller part of the common computer systems. At present, many people are working to find a niche for Forth in an ever larger and more complex world.

A Return to Diversity

In a review of the history of Forth, we see that it has been used in diverse ways. A major trend has been away from hardware and implementations. Other features have been added such as libraries. The problem with libraries is that they require a standard kernel. The advantage is that they provide the tools programmers have come to expect it as the direction that Forth seems to be taking as far as development is concerned. But there are many other uses of Forth being actively implemented in embedded processors. We need to be aware of these many embedded application where Forth can be a major contender for programmers.

A Development Platform

A Development Platform includes many tools. Not all of them are easily implemented in Forth. One of the major tools which most Forth Programmers ignore completely is version control. As applications get large with many programmers and changing requirements, it is necessary to maintain version control for maintaining programs in the future.

Then there is the problem of drivers for the many additions to hardware. Each controller seems to do things just a little differently. Separate modules for each new device can be used if the Development Platform allows linking in the various components. In such an environment, Forth becomes yet another tool. In many cases Forth is an ideal tool. This seems to be a direction that some of Forth is moving.

Embedded Systems

There are a number of simple hardware systems under computer control. These systems are dedicated to the application. There is no need for all of the over head of a full Development Platform. The Forth kernel can be very small and the dialed used is of little importance. The program need not be portable.

With such Systems we are almost back to the origins of Forth. There is a surprising number of such applications. A great diversity of Forth implementations continues to be developed.

Embedded Program Development

In many ways, it is convenient to develop the application on the system on which it is going to run. Often a simple I/O arrangement is sufficient There is no need for anything more complicated.

On the other hand, a fully equipped Development Platform can be convenient. A simulator of the embedded system can be written. The application can be target compiled for the embedded process. The tools of program control can be invoked. The embedded system can be tethered to the Development Platform and tested.

It is a choice for the programmer of the embedded system. How much time does he want to spend to learn the power of a fully endowed Development Platform, and how much time is necessary to develop the embedded application.

The answer is in diversity. We need to get beyond the Forth "choir" to the many diverse users and encourage all of their diverse ways.

A Model Operating System

Linux is an open operating system with many participants world wide. It can be very large and is certainly complex.

Some months ago, I attended a Linux Day. Actually, it was only an evening. It was attended by about 1000 people. It reminded me of the Forth Days of years past. The enthusiasm of the participants was amazing. They must be doing something right.

In thinking about what is going on, I can see many parallels to what we went through with Forth. I can see where they may well end up with some of the same problems we have in Forth.

However, for those in need of a fully equipped Development System, Linux might just provide the framework. Forth could provide another tool on a Linux platform. But, a minimal Forth as the operating system and application may be all that is necessary. I think that Forth users are scattered over the entire spectrum.

A Forth Philosophy of Diversity

It is time to open Forth to all. There is no requirement that any program is necessarily standard.

Conclusion

As Steven Pepper said in World Hypothesis, "I am dogmatically undogmatic."

SPECTRUM LABEL, NAME	ROM Addr	TS 2068 ROM Addr	TS 2068 LABEL, NAME
peek	34AC	3868	
usr-no	34B3	3872	
usr-$	34BC	38D7	
TEST-ZERO	34E9	3904	TESTO
GREATER-0	34F9	3914	
NOT	3501	391C	
less-0	3506	3921	
FP-0/1	350B	3926	STBOOL
or	351B	3936	
no-&-no	3524	393F	
str-&-no	352D	3948	
no-l-eql	353B	3956	
strs-add	359C	39B7	
STK-PNTRS	35BF	39DD	
chr$	35C9	39E4	
val-&-val$	35DE	39F9	
str$	361F	3A3A	
read-in	3645	3A60	
code	3669	3A84	
len	3674	3A8F	
dec-jr-nz	367A	3A95	
JUMP	3686	3AA1	
jump-true	368F	3AAA	
end-calc	369B	3AB6	
n-mod-m	36A0	3ABB	INTDIV
****	****	3AC5	LDMES
int	36AF	3ACA	INT
EXP	36C4	3ADF	EXP
ln	3713	3B2E	
get-argt	3783	3B9E	ANGLE
cos	37AA	3BC5	COS
sin	37B5	3BD0	SIN
tan	37DA	3BF5	TAN
atn	37E2	3BFD	ATN
asn	3833	3C4E	ASN
acs	3843	3C5E	ACS
sqr	384A	3C65	ROOT
to-power	3851	3C6C	TO-THE
****	****	3C89	SEPRMT
(note:SEPRMT are tape msgs.)			
'spare'	386E	3CDC	'EMPTY'
charctr-set	3D00	3D00	CH-SET

The Spectrum does not support the following routines:

****	****	17B5	AROS
****	****	17CF	GETAL
****	****	17EA	AR-LN
****	****	17FF	AR-NXT
****	****	18C6	AAROS

TIMEX EXROM

SPECTRUM LABEL, NAME	ROM Addr	TS 2068 ROM Addr	TS 2068 LABEL, NAME
SA-BYTES	04C2	0068	W-TAPE
SA-LEADER	04D8	007E	
SA-LOOP	04FE	00A4	
SA-BIT-2	0511	00B7	
SA-8-BITS	0525	00CB	
SA/LD-RET	053F	00E5	W-BORD
REPORT-D	0552	00F8	
LD-BYTES	0556	00FC	R-TAPE
LD-BREAK	056B	0111	
LD-LEADER	0580	0126	
LD-SYNC	058F	0135	
LD-8-BITS	05CA	0170	
LD-EDGE-2	05E3	0189	RD-BIT
LD-EDGE-1	05E7	018D	RD-EDGE
LD-SAMPLE	05ED	0193	
SAVE-ETC	0605	01AB	SLVM
REPORT-F	0642	0228	
SA-NAME	064B	0231	
SA-DATA	0652	0238	
SA-V-OLD	0672	029A	
SA-V-NEW	0685	02A9	
SA-SCR$	06A0	02F2	
SA-CODE	06C3	032E	
SA-LINE	0716	0447	
SA-ALL	075A	04C9	
LD-LOOK-H	0767	04D6	
LD-NAME	07A6	053D	
VR-CONTROL	07CB	058F	VERIFY
LD-BLOCK	0802	05C6	
LD-CONTROL	0808	05CC	LOAD
LD-DATA	082E	0606	
LD-PROG	0873	0673	
ME-CONTROL	08B6	06E5	MERGE
ME-OLD-VP	08F9	0752	
ME-ENTER	092C	0799	
ME-ENT-1	093E	07CF	
ME-ENT-3	0958	0825	
SA-CONTROL	0970	0851	SAVE
SA-1-SEC	0991	089A	

The balance of EXROM contains the Function Dispatcher , Bank Switching Code , and various other routines , which does not have counterparts in the Spectrum. A total of approximately 2K Bytes of EXROM is unused.

SPECTRUM LABEL, NAME	ROM Addr	TS 2068 ROM Addr	TS 2068 LABEL, NAME
INDEXER	16DC	1368	SEARCH
		1374	SRCHSC
CLOSE	16E5	139F	CLOSE
	16EB	13A8	RSTSTR
CLOSE-2	1701	13BE	CLCHAN
OPEN	1736	142A	OPEN
OPEN-1	1756	145E	
OPEN-2	175D	1465	OPCHAN
CAT-ETC.	1793	25C8	CAT
AUTO-LIST	1795	14E1	LIST
LLIST	17F5	1541	K-LLST
LIST	17F9	1545	K-LIST
	1860	15AC	LPO
OUT-LINE	1855	15A1	PUT-SR?
OUT-LINE2	187D	15C9	PUT
NUMBER	18B6	1602	
OUT-FLASH	18C1	160D	FLASHA
OUT-CURS	18E1	162D	PR-CUR
LN-FETCH	190F	165B	NEXT-L
LN-STORE	191C	1668	DE-HL
OUT-SP2	1925	1671	
LINE-ADDR	196E	16D6	FIND-L
CP-LINES	1980	16E8	CP-BC
	1988	16F0	SUBLIN
EACH-STMT	198B	16F3	SUBLN1
NEXT-ONE	19B8	1720	RECLEN
DIFFER	19DD	1745	
RECLAIM-1	19E5	174D	DEL-DE
RECLAIM-2	19E8	1750	DELREC
E-LINE-NO	19DF	1768	LINENO
OUT-NUM-1	1A1B	1788	PUT-BC
OUT-NUM-2	1A28	1795	PU-LN
OUT-NUM-3	1A30	179D	
LINE-SCAN	1B17	1A27	SYNTAX
STMT-LOOP	1B28	1A44	LS4
SEPARATOR	1B6F	1AB2	
STMT-RET	1B76	1AB9	
LINE-RUN	1B8A	1AD8	EXCUTE
LINE-NEW	1AEC	1B9E	
REM	1BB2	1B00	
LINE-END	1BB3	1B09	
LINE-USE	1BBF	1B15	
NEXT-LINE	1BD1	1B27	
STMT-RET	1B76	1AB9	ENDSTT
CHECK-END	1Bee	1B44	END?
STMT-NEXT	1BF4	1B4A	ENDTEM
CLASS-01	1C1F	1B82	TEM1
REPORT-2	1C2E	1B91	ERR2
VAL-FET-2	1C59	1B8C	LT22

SPECTRUM LABEL, NAME	ROM Addr	TS 2068 ROM Addr	TS 2068 LABEL, NAME
NEXT-2-NUM	1C79	1BDC	DYADIC
CLASS-06	1C82	1BE5	TEM6
REPORT-C	1C8A	1BED	SYNERR
CLASS-0A	1C8C	1BEF	TEM10
PERMS	1C96	1BF9	
FETCH-NUM	1CDE	1C49	OPTNO
USE-ZERO	1CE6	1C51	STK-0
STOP	1CEE	1C59	STOP
IF	1CF0	1C5B	
FOR	1D03	1C78	FOR
LOOK-PROG	1D86	1D28	SKIP
NEXT	1DAB	1D55	NEXT
READ	1DEC	1D96	READ
DATA	1E27	1E82	DATA
RESTORE	1E42	1E9D	
RANDOMIZE	1E4F	1ED4	RAND
REST-RUN	1E45	1ECA	RESTBC
CONTINUE	1E5F	1EE4	CONT
GO TO	1E67	1EF1	JUMP
OUT	1E7A	1F04	
POKE	1E80	1F0A	
TWO-PARAM	1E85	1F0F	
FIND-INT1	1E94	1F1E	FIX-U1
FIND-INT2	1E99	1F23	FIX-U
REPORT-B	1E9F	1F29	ERRB
RUN	1EA1	1F2E	
CLEAR	1EAC	1F36	CLEAR
CLEAR-RUN	1EAF	1F39	CLR-BC
GOSUB	1EED	1F99	GO-SUB
TEST-ROOM	1F05	1FBB	CHK-SZ
REPORT-4	1F15	1FCF	ERR4
RETURN	1F23	1FD4	RETURN
PAUSE	1F3A	1FEF	PAUSE
BREAK-KEY	1F54	2009	BREAK
DEF FN	1F60	201D	DEF
****	****	2128	SOUND
UNSTACK-Z	1FC3	214F	
LPRINT	1FC9	2155	K-LPR
PRINT	1FCD	2159	K-PRN
PRINT-2	1FDF	217E	P-SEQ
PRINT-CR	1FF5	2194	
PRINT-ITEM1	1FFC	219B	
PR-STRING	203C	21DB	
PR-END-Z	2045	21E4	
PR-ST-END	2048	21E7	TERM?
PR-POS-1	204E	21ED	
STR-ALTER	2070	220F	STRITO
INPUT	2089	222B	INPUT
IN-ITEM-1	20C1	226B	I-SEQ

SPECTRUM LABEL, NAME	SPECTRUM ROM Addr	TS 2068 ROM Addr	TS 2068 LABEL, NAME
IN-ASSIGN	21B9	2363	
REPORT-H	21D4	237E	ERRH
IN-CHAN-K	21D6	2380	NOTKB?
CO-TEMP-1	21E1	2388	
CO-TEMP-2	21E2	238C	GR-COL
CO-TEMP-4	21FC	23A6	COLITM
CO-TEMP-7	2234	23DE	COLOUR
CO-CHANGE	226C	2416	
CO-TEMP-C	2273	241D	HIFLSH
BORDER	2294	243E	BORDER
****	****	24D2	NEWDEV
****	****	25B9	PASSEM
PIXEL-ADD	22AA	2603	SCRMBL
POINT	22CB	2624	F-PNT
PLOT	22DC	2635	PLOT
PLOT-SUB	22E5	263E	PLOTBC
STK-TO-BC	2307	2660	GETXY
STK-TO-A	2314	266D	
CIRCLE	2320	2679	CIRCLE
DRAW	2382	26DB	DRAW
CD-PRMS1	247D	27D6	
DRAW-LINE	24B7	2810	DRAW-L
SCANNING	24FB	2854	EXPRN
SYNTAX-Z	2530	2889	INTPT?
S-ATTR-s	2580	28D7	F-ATTR
S-U-PLUS	25AF	296D	
S-LETTER	26C9	2A87	
S-FN-SBRN	27BD	2B7B	
S-SCREEN$-S	2535	288E	F-SCRN
S-RND	25F8	29B6	RND
S-PI	2627	29E5	F-PI
S-INKEY$	2634	29F2	F-INKEY
FN-SKPOVER	28AB	2C69	NXT-HL
LOOK-VARS	28B2	2C70	FIND-N
STK-F-ARG	2951	2D0F	
STK-VAR	2996	2D54	GET-EL
SLICING	2A52	2E10	SLICER
STK-ST-0	2AB1	2E6F	
STK-STO-1	2AB2	2E70	PSHSTR
STK-STORE	2AB6	2E74	PAEDCB
INT-EXP-1	2ACC	2E8A	
DE,(DE+1)	2AEE	2EAC	
GET-HL*DE	2AF4	2EB2	
LET	2AFF	2EBD	LET
L-ENTER	2BA6	2F64	
L-ADD-$	2BAF	2F6D	
L-STRING	2BC6	2F84	
L-FIRST	2BEA	2FA8	
STK-FETCH	2BF1	2FAF	POPSTR
DIM	2C02	2FC0	DIM

SPECTRUM LABEL, NAME	SPECTRUM ROM Addr	TS 2068 ROM Addr	TS 2068 LABEL, NAME
ALPHANUM	2C88	3046	ALNUM?
ALPHA	2C8D	304B	ALPHA?
DEC-TO-FP	2C9B	3059	STKSUM
NUMERIC	2D1B	30D9	DIGIT?
STK-DIGIT	2D22	30E0	
STACK-A	2D28	30E6	STK-A
STACK-BC	2D2B	30E9	STK-BC
INT-TO-FP	2D3B	30F9	ININT
E-TO-FP	2D4F	3100	
INT-FETCH	2D7F	313D	LDDE
P-INT-STO	2D8C	314A	STDE-U
INT-STORE	2D8E	314C	STDE-S
FP-TO-BC	2DA2	3160	FP2BC
LOG(2+A)	2DC1	317F	
FP-TO-A	2DD5	3193	FP2A
PRINT-FP	2DE3	31A1	OUTPUT
CA=10*A+C	2F8B	334A	
PREP-ADD	2F9B	335A	
FETCH-TWO	2FBA	3379	SUMSLD
SHIFT-FP	2Fdd	339C	SHIFT
ADD-BACK	3004	33C3	
SUBTRACT	300F	33CE	SUB
addition	3014	33D3	ADD
HL=HL*DE	30A9	3468	MULT
PREP-M/D	30C0	347F	
multiply	30CA	3489	
REPORT-6	31AD	356C	ERR6
division	31AF	356E	DIVIDE
truncate	3214	35D3	TRUNC
RE-ST-TWO	3293	3052	
multiply	30CA	3489	TIMES
RE-STACK	3297	3656	FLOAT
FP calculator start:			
	32C5	3684	
CALCULATE	335B	371A	CTRO
fp-calc-2	33A2	3761	
TEST-5-SP	33a9	3768	ROOM?
STACK-NUM	33B4	3773	STK-M
MOVE-FP	33C0	377F	RAMNO
STK-DATA	33C6	3785	
SKIP-CONS	33F7	37B6	
LOC-MEM	3406	37C5	ARRAY
get-mem-0	340F	37CF	
stk-zero	341B	37DA	
st-mem-0	342D	37EC	
EXCHANGE	343C	37FB	
series-06	346A	3829	
NEGATE	346E	382D	NEGATE
sgn	3492	3851	
in	34A5	3864	

SPECTRUM LABEL, NAME	SPECTRUM ROM Addr	TS 2068 ROM Addr	TS 2068 LABEL, NAME
START	0000	0000	PLUGIN
ERROR-1	0008	0008	
PRINT-A-1	0010	0010	WRCH
GET-CHAR	0018	0018	
NEXT-CHAR	0020	0020	
FP-CALC	0028	0028	
BC-SPACES	0030	0030	
MASK-INT	0038	0038	
	004F	004F	PHLAF
ERROR-2	0053	0053	
ERROR-3	0055	0055	LE3
RESET	0066	0066	
NO-RESET	0070	0070	
CH-ADD+1	0074	0074	NEXTCH
TEMP-PTR1	0077	0077	NC-HL
TEMP-PTR2	0078	0078	TC-HL
SKIP-OVER	007D	007D	
TOKENS	0095	0098	TOKENS
KEYTBL	0227	0245	
KEY-SCAN	028E	02B0	K-SCAN
KEYBOARD	02BF	02E1	UPD-K
K-REPEAT	0310	0336	
K-TEST	031E	035C	K-BASE
K-DECODE	0333	0371	CHCODE
BEEPER	03B5	03F3	PARP
BEEP	03F8	0436	BEEP
PRINT-OUT	09F4	0500	SEND-TV
PO-BACK-1	0A23	053A	P-LFT
PO-RIGHT	0A3D	0554	P-RT
PO-ENTER	0A4F	0566	P-NL
PO-COMMA	0A5F	0576	
PO-QUEST	0A69	0580	
PO-TV-2	0A6D	0584	
	0A9B	05B2	SET-AT
PO-ABLE	0AD9	05F0	
PO-STORE	0ADC	05F3	STTVCU
PO-FETCH	0B03	061A	LDTVCU
PO-ANY	0B24	0638	
PO-ALL	0B7F	0684	
PO-ATTR	0BDB	0710	ATTBYT
PO-MSG	0C0A	073F	PUTMES
PO-SAVE	0C3B	0776	PR-TV2
PO-SEARCH	0C41	077C	
PO-SCR	0C55	0790	TVFUL?
REPORT-5	0C86	07C1	ERR5
TEMPS	0D4D	0888	R-ATTS
CLS	0D6B	08A6	K-CLS
CLS-LOWER	0D6E	08A9	CLLHS
CL-ALL	0DAF	08EA	CLS
CL-SET	0DD9	0914	SET-CUR
CL-SC-ALL	0DFE	0939	SCRL
CL-LINE	0E44	097F	CLS-B

SPECTRUM LABEL, NAME	SPECTRUM ROM Addr	TS 2068 ROM Addr	TS 2068 LABEL, NAME
CL-ATR	0E88	09C3	
CL-ADDR	0E9B	09D6	
COPY	0EAC	0A02	K-DUMP
COPY-BUFF	0ECD	0A23	DUMPPR
CLEAR-PRB	0EDF	0A35	CLPR
COPY-LINE	0EF4	0A4A	PRSCAN
EDITOR	0F2C	0A82	EDIT-K
ADD-CHAR	0F81	0AE7	INSA
ED-EDIT	0FA9	0B12	
ED-DOWN	0FF3	0B59	
ED-LEFT	1007	0B6B	
ED-RIGHT	100C	0B73	
ED-DELETE	1015	0B7B	DELSYM
ED-ENTER	1024	0B8A	
ED-EDGE	1031	0B97	
ED-UP	1059	0B8F	
ED-SYMBOL	1076	0BDC	
ED-ERROR	107F	0BE9	
CLEAR-SP	1097	0BFE	DEL-K
KEY-INPUT	10A8	0C0E	IN-K
ED-COPY	111D	0C83	ECHO
REMOVE-FP	11A7	0D0D	DESLUG
NEW	11B7	0D1D	K-NEW
START/NEW	11CB	0D31	INIT
RAM-SET	1219	0D7F	NEW
MAIN-EXEC	12A2	0E28	
MAIN-1	12A9	0E2F	LED18
MAIN-4	1303	0ED8	LED4
REPORT-MSG	1391	0F65	
MAIN-ADD	155D	1158	
CH-INFO	15AF	11AA	CHINIT
INIT-STR	15C6	11C1	SHINIT
WAIT-KEY	15D4	11CF	RDCH
INPUT-AD	15E6	11E1	INCH
OUT-CODE	15EF	11EA	PUTDIC
PRINT-A-2	15F2	11ED	SENDCH
CHAN-OPEN	1601	1230	SELECT
REPORT-O	160E	123D	ERRO
CHAN-FLAG	1615	1248	SEL-HL
CHAN-K	1634	129A	
CHAN-s	1642	12A8	
CHAN-P	164D	12B3	
ONE-SPACE	1652	12B8	INS1
MAKE-ROOM	1655	12BB	INSERT
POINTERS	1664	12CA	REMGSZ
LINE-ZERO	169F	131E	
LINE-NO	1695	1324	GET-LN
RESERVE	169E	132D	LCU2
SET-MIN	16B0	133F	CLEL
SET-WORK	16BF	134E	X-CALC
REC-EDIT	16D4	1363	X-T-HL
SET-STK	16C5	1354	RESET

EZ80 MICRO COMPUTER©

WilfRigter

Introduction 1 Jan, 1998 revision

The EZ80 (pronounced "eazy 80") is an all CMOS Z80 controller with some novel features. The simple design, powerful debugging features and low power (<10 ma) consumption make it suitable for battery powered robotics hand-held and/or educational applications. The EZ80 consists of a CMOS Z80, 32K of battery backed up CMOS RAM, a CMOS 8255 with 24 I/O lines, and five simple 74HC type glue chips. The EZ80 connects to the world via three connectors: the EZiCE port, the EZI/O port and the EZKEY port.

No EPROM?", you say? The core features of the EZ80 are the BLANK memory and the unique on-board programming hardware which does not need a monitor program in EPROM. This makes the EZ80 similar to some single chip micro-controller chips with on-chip EEPROM programming capability. The programming functions operate on single bytes or blocks of up to 32K bytes. Program code is normally stored in the 0-16K block which is non-volatile and write protected. The read/write data is stored in the 16K-32K block. Actually programming is only a small part of the build-in debugging hardware called the EZiCE port.

During program and hardware development the EZiCE port is connected to a standard IBM PC parallel port. The EZ80 single stepper circuit, when combined with a control decoder and 3-state buffer can transfer opcodes and data between the PC printer port and the Z80 data bus and ultimately the Z80 registers. This method of "stuffing" opcodes on the Z80 data bus is also used to generate random access addressing to program the on board non-volatile memory and perform a range of debugging functions. The EZiCE software so far provides the following menu:

```
1. FILE          8. BREAKPOINT
2. LOAD          9. RUN
3. SAVE         10. RESET
4. VERIFY       11. SETUP
5. LIST         12. SHELL
6. STEP         13. USER
7. TRACE        14. EXIT
```

The EZ80 uses a 8255 24-bit parallel interface with four registers corresponding to 3 general purpose 8-bit TTL computable I/O ports and a control register. The 8255 supports the EZI/O port and the EZKEY port.

The EZI/O port emulates a PC printer port to take advantage of the many LPT peripheral projects designed for the PC. Conversely any EZ80 I/O expansion project should be computable with the PC LPT port as well!

The EZKEY port is connected to 8255 Port B and has added hardware to interface directly with a scanned 64 contact keyboard matrix. The 26 pin EZKEY connector is also used for other front panel functions such as the POWER and RESET switches and 5 LEDs.

The EZiCE port is a Direct Memory Access (DMA) interface designed to connect to a PC LPT printer port during programming and debugging. In the RUN mode, the EZiCE HC573 latch has a secondary function as a general purpose TTL level 8-bit input register.

Circuit Overview

The EZ80 circuit details are shown in the schematic fig 1. The basic design is simply a Z80 connected to a 32K SRAM and some I/O. The simple battery backup for the SRAM is connected to the RAM Vss pin and uses 5V as the common. This has a big advantage over the more conventional designs in that the WR and CE lines are automatically pulled up to Vcc during power down. The RAM is partially write protected providing 16K bytes of write protected memory from 0K-16K and 16K of read/write memory from 16K-32K.

The RESET circuit uses a push button, a decoded reset command and a power supply under-voltage monitor to reset the Z80 and 8255 chips. This prevents undefined Z80 operations during power cycling which could corrupt memory or send spurious signals to the I/O pins.

The basic I/O circuit consists of a 24-bit I/O 8255 PPI chip and an 8-bit tristate buffer which connect the EZ80 to the outside world via the EZI/O port, the EZKEY port and the EZiCE port.

The EZiCE port is a on-board In Circuit Emulation interface used for program and hardware development. A key component is the SINGLE STEPPER circuit to step the CPU one bus cycle at a time. The CPU uses bus cycles to access memory or I/O devices which can be a M1 opcode fetch or any other read or write cycle. The EZiCE PORT uses the PC LPT control lines to override the Z80 IORQ, RD and WR control signals. When these signals are applied in the correct sequence, the EZiCE port provides a kind of direct memory access (DMA) to the EZ80 RAM and Z80 data bus. Data from RAM and CPU bus (and CPU registers) is read in two nibbles to four of LPT status lines. The fifth line (ERR) conveys M1 and HALT.

The EZI/O PORT uses the 8255 Port A and Port C to emulate a true PC type bi-directional printer port.

The EZKEY PORT uses the 8255 Port B and the A8-A15 CPU address line to scan a matrix of up to 64 key contacts.

Z80 Inside

The EZ80 uses a CMOS Z80 CPU for reduced power consumption. Details of the CPU functions can be found in the ZILOG Z80 hand book. The section on bus cycles requires close examination since during "single stepping" each bus cycle is frozen and the data can be examined.

The Z80 is reset both from a switch, a TL7705A supervisory circuit which pulls the reset line low when Vdd falls below 4.75V and reset is also controlled via the EZiCE port. ZX-81 fans will be happy to hear that with minor modifications, the ZX-81 ROM code is compatible with the EZ80 hardware and control programs can be written in ZX-81 BASIC. Without ZX video, the INT and NMI lines are now available on the board for user applications. Future projects will customize the ZX-81 ROM for use with the EZ80 I/O and such add

on hardware as a DFILE computable LCD interface.
What makes it tick?

The 4 MHz CPU clock is generated with a text book low power CMOS Pierce oscillator using one section of the 74HC02 NOR gate as an inverter with the crystal in the feedback path.

Lest we forget

The 62256 RAM chip, which is also low power CMOS, uses a 3V lithium cell for non-volatile program storage. Note the novel use of a common Vdd line and the negative battery connection to the RAM Vss line. This ensures that the RAM CE and WR lines are disabled by automatically switching to Vdd during powerdown. With the simple one chip memory, no address decoding is required for the EZ80. However the WRITE PROTECTOR circuit divides the memory map as follows: 16K of program memory from 0K-16K is write protected during normal program execution and the 16K-32K block is always accessible as read/ write RAM. All memory is read/write accessible in the programming mode.

Boot Strap?

The EZ80 uses a hardware "boot strap" circuit to program the "blank" memory from power up. This is similar to many modern single chip microcontrollers. Ironically, a similar procedure was used in the early days of ROMless computers when programs were entered manually through the front panel switches. That toggling procedure was tedious and only a short "boot strap" program was loaded through the front panel which was just enough to start the paper tape reader to load in the next layer of operating system software. The way a PC initializes it's BIOS and DOS in successively more complex layers follows a similar sequence but has evolved so that by now the operator's front panel actions have been reduced to, at most, using in the proverbial three finger salute! In fact a bank of 12 switches and some LEDs could still be used to program the EZ80 but instead we have finally found a useful application for the IBM PC, as a high priced front panel for the EZ80.

EZ Ports Rule, OK!

The EZ80 circuit in Fig 1 shows EZI/O port using a 26 pin male header which matches a standard IDC DB25 connector. With this, the EZ80 can be connected to a standard printer or other PC printer port compatible device. It can also be connected to the PC printer port for high speed bi-directional data transfer. Future EZI/O expansion devices also use this bus for data transfer but will be connected via a 26 pin connector and flat cable. Note the use of pin 26 to distribute +5V to EZI/O expansion devices which can also be powered from a separate external power supply. The EZI/O port is accessed at I/O address 00.

The EZiCE port uses a 74HC541 tristate buffer, a 74HC257 multiplexer, a 74HC02 NOR gate, a 74HC74 D flip flop and a 74HC139 decoder. The 74HC139 is used to decode the PC printer port control lines during programming to control and override various internal EZ80 signals. The 74HC139 control functions and codes are shown in fig 7 for the relevant PC printer port control line states. The LPT control register codes are used to reset, single step and run the Z80 and to control the RAM OE and WR lines. The 74HC541 buffer is connected to the PC printer port data lines which are used to control the EZ80 CPU data bus. The tristate buffer is controlled by the PC to write data from the LPT data lines to the CPU data bus. It is used by the PC to "stuff" opcodes and data to the Z80 register or to write data to the RAM.

The 74HC257 is used to read the memory data in two nibbles to 4 of the PC printer port status lines. The PC LPT ERR status line is connected to the Z80 HALT and M1 lines to monitor and trap Breakpoints and M1 opcode fetch cycles during debugging. In the RUN mode, the EZ programmer port can be read as a general purpose 8-bit input port at I/O address 04.
The EZKEY port is similar to and compatible with the ZX-81 keyboard port.

This keyboard interface connects up to 64 contacts of a matrix keyboard. It uses diode isolated A8-15 lines to scan 8 rows and 8-bit of column data are read on Port B at I/O address 01. In order to use the address lines for scanning the keyboard rows we can use the Z80 IN A,(C) opcode to scan each row while rotating a low-bit through the B register. This opcode uses the C register to address the I/O port on the A0-A7 lines while the B register appears simultaneously on the A8-A15 lines. If fewer keys are required (i.e. 40 keys for ZX-81 emulation), only 5-bits of Port B are needed while the other three-bits (D5-7) can be used for a synchronous serial interface or for a-bit banger UART application.

One Small Step for Man

The novel 74HC74 SINGLE STEPPER circuit controls the Z80 WAIT line to single step the Z80 one bus cycle at a time. The WAIT line is normally used to lengthen CPU bus cycles in order to synchronize the CPU bus to devices with slow access time. In the case of the single stepper, the CPU is held in the WAIT state and is advanced one bus cycle at a time by "tugging" the WAIT line with a "step" pulse. When the single stepper is combined with the 74HC541 and the 74HC139 control decoder, the PC can force external OPCODES on the Z80 data bus during each instruction cycle while the RAM OE line is disabled. The RUN signal activates the 74HC74 preset input to force the WAIT line positive and to allow the CPU to run at full speed. The WAIT line is also used to enable the NIBBLER circuit to read the CPU data bus to the LPT status register one nibble at a time. I really like this simple Z80 single stepper circuit which is destined to become a classic (if it isn't already). It adapts itself to any CPU clock frequency unlike the usual monostable pulse generator designs. The single stepper timing states are shown in fig 3.

One Giant Leap for Mankind

In the programming mode, the EZiCE software uses combinations of control codes to enable the 74HC541 to place externally generated disabling the RAM Output Enable (OE) line. This can be used to "stuff" opcodes on the CPU data bus. This is such a powerful feature that I have only uncovered the tip of the iceberg of possibilities. For example, the PC software presently stuffs NOPs to the CPU to increment the Z80 program counter (and address lines) to desired starting address but it is just as easy and a lot faster to write a 3 byte jump instruction to load the PC counter. NOPs are still used to STEP through and LIST a block of the memory contents. Ideally, the data read by LIST is automatically disassembled to mnemonics.

(This one is on my wish list :) During TRACE, the single stepper simply advances the Z80 to the next bus cycle without forcing data on the CPU bus. The PC can use ERR status line to advance as many bus cycles to get to the next M1 cycle. At each M1 cycle the contents of the Z80 registers can be examined with a couple of POP and PUSH instructions. As you can see this is pretty powerful stuff for such a simple circuit! While single stepping through a program in memory , the PC writes a sequence of opcodes at each M1 cycle which makes the Z80 describe it's current register state and then restores everything back before executing the next memory instruction.

The current revision of the PC software does not yet have capability to perform this task in a transparent fashion. Any volunteers? The RUN command simply releases the WAIT line and allows the CPU to execute the code in real time from a given starting address. This is useful to check out subroutines, etc. In order to return control to the PC after executing the subroutine or any other code segment we want to be able to insert a break point at the point where the PC takes over. This can be done by inserting a HALT in the code e.g. replacing the RET instruction at the end of the subroutine. The Z80 HALT line is connected to the PC LPT ERR status line (and the M1 line is disconnected at this time) and when the HALT instruction is executed, the CPU pulls the HALT line low and stops. When HALT is active, the PC dumps the Z80 registers including the Program Counter (PC) and compares the PC to a list of break point addresses and then restores the missing instruction. The current PC software only has a rudimentary version of this installed.

Yippee I/O

The 8255 is a general purpose parallel I/O port. The 8255 defaults to 3 input ports on reset and to configure ports for output, a control word must be written to the control register. Port A and C are used for the EZI/O PORT, a bi-directional PC computable printer port. Port B is used as a 64 contact keyboard interface. The 8255 can also be used in small applications 24-bit TTL parallel I/O. The 8255 uses the A and B pins, connected A0 and A1 to address internal registers. The 8255 chip select is connected to a 74HC139 and is enabled when A2 is low. The 8255 data registers for ports A, B and C and control registers are addressed at 00, 01, 02 and 03 respectively. Note: Port C output-bits can be accessed at two different locations. Port C is accessed at address 02 but in addition, individual Port C-bits can be set or reset by writing data values 00 to 0F to the control register at address 03. Use even data, 02 to 0E, to reset-bits 0 to 7 respectively and odd data 01 to 0F set-bits 0 to 7. In a future article, applications will be presented which will explore all the features of this excellent chip. In the RUN mode the 74HC541 can be used as a general purpose 8-bit input port located at I/O address 04.

The EZI/O port uses 8255 Port A, a true bi-directional data port, for data lines and Port C for the 4 CONTROL and 4 STATUS lines. The EZI/O connector matches the PC LPT pinouts except for the ERR line, and connects via a 26 pin male header using standard PC DB25 connecting cables to standard LPT devices like a printer or PC LPT.

Several EZ80 digital and analog I/O expansion units, which we will design in future articles are, can be connected in parallel to the EZI/O CON which also provides +5V power on pin 26. As a bonus, these I/O expansion units can be used with any IBM PC.

Who Controls the Controller?

Fig 6 shows how PC LPT control lines are used to control access to the EZ80 internal bus. Plugging in the PC cable enables programming mode and the appropriate LPT control codes can be used to read, write, etc. Fig 7 shows the state of the LPT control lines as well as the internal control signals and the control code which is written to the PC LPT control register. This control code value is made up from binary weighted values (shown in brackets in fig 6) but some register-bits are inverted with respect to the control line. Therefore writing a 0 to some control register-bits will result in a 1 state of the control line. The STB (1), ALF (2), and SEL (8)-bits are all inverted but the INI (4) line is true. In fig 7, a plus or minus indicates a true or inverted-bit.

No wonder the odd relationship between the control code and the state of the control lines. Note that WR pulses have to be generated by writing three control codes in sequence. Similarly, writing OPCODES and TRACING bus cycles require a sequence of control codes. I have shown one possible sequence but other sequences which combine stepping and reading the data bus are also possible. It must be remembered that DATA must be stable at certain times and the STB line must be held high while changing the INI and ALF lines to avoid glitching of the decoder outputs. This is important because of the slow rise and fall times of the LPT port signals. Note the SEL line, which activates the single stepper, is conditioned with a Schmidt trigger to avoid double stepping. The INI lines is also used to enable the RAM OE or the 74HC541. When the EZiCE port is disconnected from the PC LPT, the control lines are pulled up and automatically decodes the RUN control signal and thereby enables the EZ80 RUN mode. While in the RUN mode, the 74HC541 can be used as an input port but the control lines must be left disconnected.

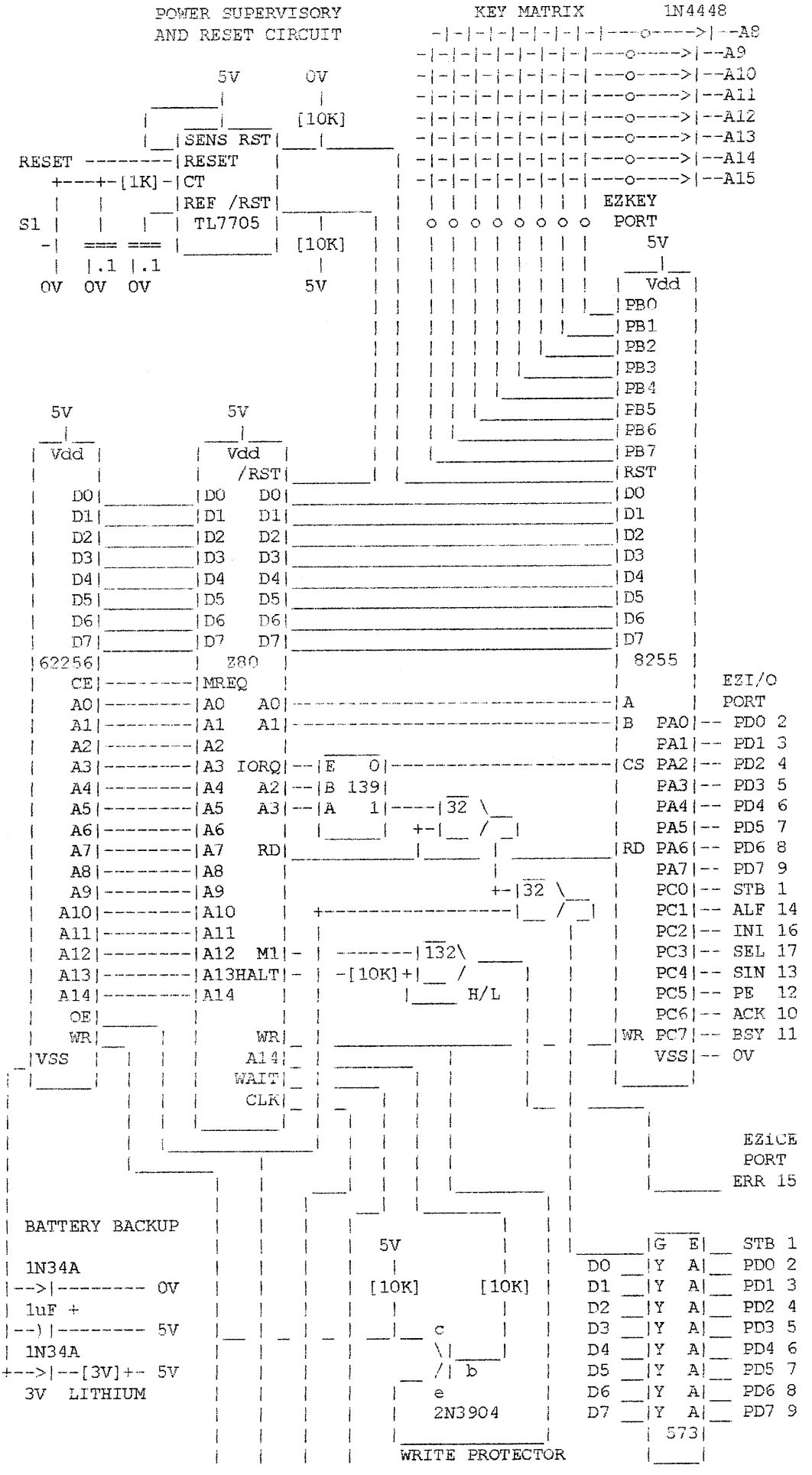
Wrap Up

What can I say about the EZ80 that I haven't already said twice? I was pleasantly surprised to discover a number of new ideas/features just writing the article. That's always a good indication that there are more features waiting to be uncovered. This project feeds my belief that many interesting things happen at the simple end of the complexity spectrum. This is the nurture phase of the EZ80 project where I look for some support. The simple hardware, elegant design and interesting attributes make the EZ80 "eazy" to use for a variety of applications. Let's have those ideas, comments, corrections and let's have some fun! Use the mail server for group discussion but feel free to contact me at: rigter@cafe.net

Revision History

27 Dec, 1997 - released preliminary design.

1 Jan, 1998 - Added 74HC541, revised control codes, replaced 74HC138 with 74HC139, replaced MC32164 with TL7705, changed I/O addresses, reduced number of passive components, removed run/prog switch, changed port descriptions, refined article layout. THANK YOU! To Clive Sinclair's "small is beautiful" ZX-81. To all who have commented with helpful criticism and kind remarks. Enjoy. **Wilf**


POWER SUPERVISORY
AND RESET CIRCUIT
KEY MATRIX
1N4448
5V
0V
[10K]
SENS RST
RESET
CT
REF /RST
TL7705
[1K]
S1
.1
.1
EZKEY
PORT
5V
Vdd
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
RST
62256
Z80
MREQ
IORQ
M1
HALT
WR
RD
A14
WAIT
CLK
8255
EZI/O
PORT
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
STB 1
ALF 14
INI 16
SEL 17
SIN 13
PE 12
ACK 10
BSY 11
VSS
0V
EZ1CE
PORT
ERR 15
BATTERY BACKUP
1N34A
1uF
3V LITHIUM
2N3904
WRITE PROTECTOR
573
PD0 2
PD1 3
PD2 4
PD3 5
PD4 6
PD5 7
PD6 8
PD7 9

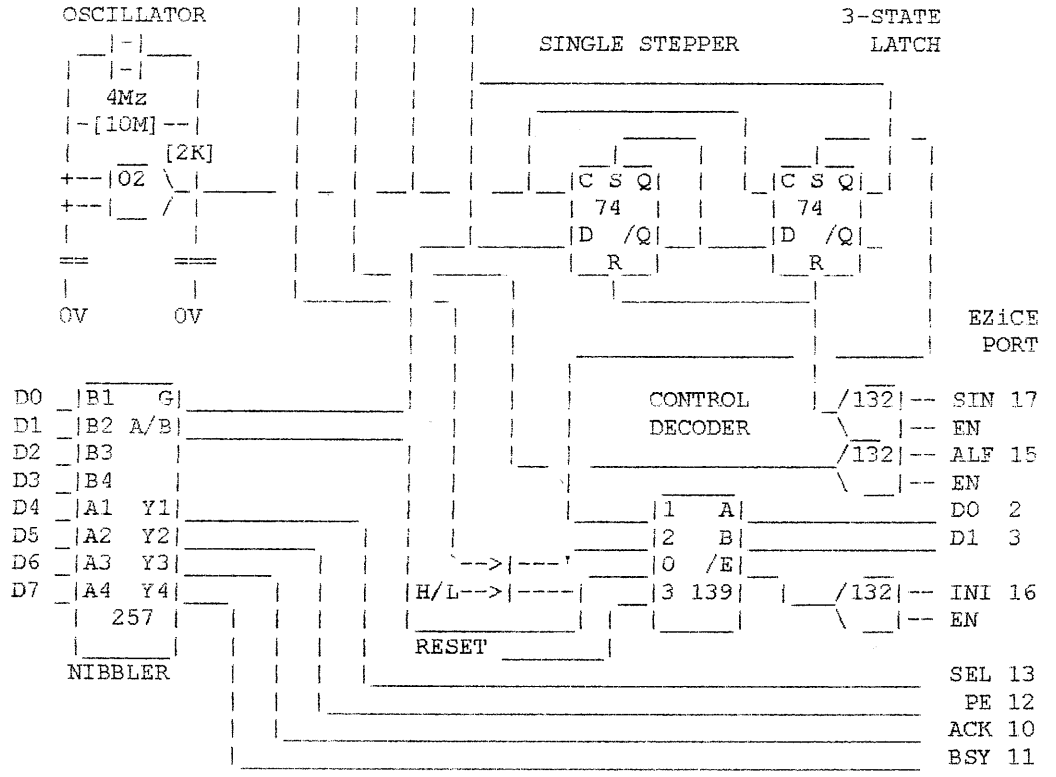


Figure 1 - The EZ80 Complete Circuit

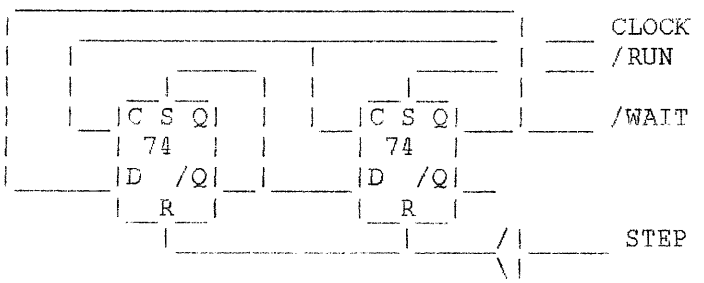


Figure 2 - The Single Stepper

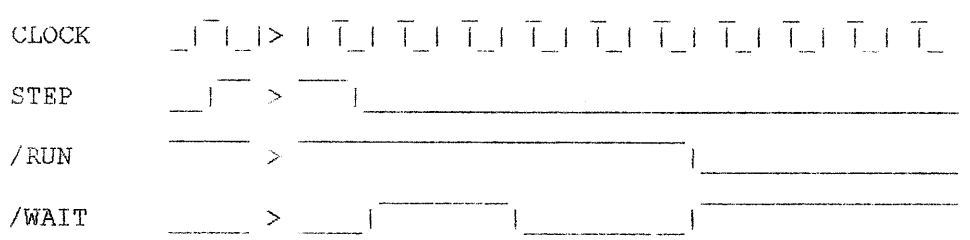


Figure 3 - Single Stepper Timming

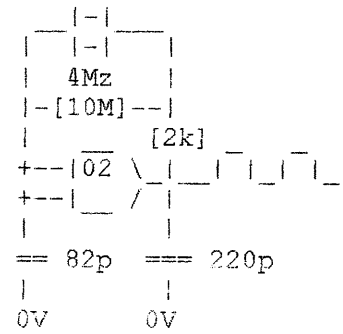


Figure 4 - Oscillator

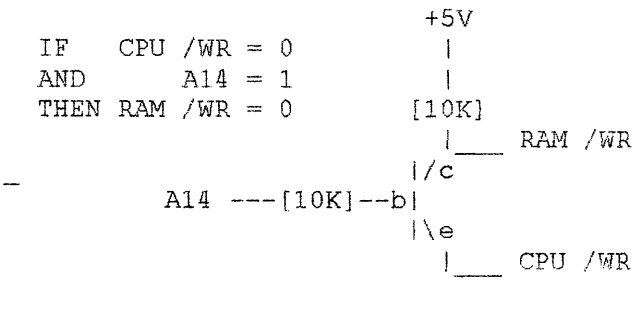


Figure 5 - Write Protector

Control Decoder

CONTROL FUNCTION	CONTROL CODE	<--LPT CONTROL--> -STB	-ALF	+INI	-SEL	LPT DATA	<---EZ80 CONTROL SIGNALS---> WAIT	OE	H/L	WR	RST	RUN
READ RAM LOW*	01	0	1	0	1	X	0	0	1	1	1	1
READ RAM HIGH	00	1	1	0	1	X	0	0	0	1	1	1
WRITE MEMORY(1)	07	0	0	1	1	X	0	1	1	1	1	1
WRITE MEMORY(2)	06	1	0	1	1	STABLE	0	1	1	0	1	1
WRITE MEMORY(3)	07	0	0	1	1	STABLE	0	1	1	1	1	1
WRITE OPCODE(1)	03	0	0	0	1	X	0	1	1	1	1	1
WRITE OPCODE(2)	11	0	0	0	0	STABLE	0	1	1	1	1	1
WRITE OPCODE(3)	03	0	0	0	1	STABLE	1*	1	1	1	1	1
TRACE OPCODE(1)	01	0	1	0	1	X	0	0	1	1	1	1
TRACE OPCODE(2)	09	0	1	0	0	X	0	0	1	1	1	1
TRACE OPCODE(3)	01	0	1	0	1	X	1*	0	1	1	1	1
RESET	02	1	0	1	1	X	0	1	1	1	0	1
RUN*	06	1	0	0	1	X	1	1	1	1	1	0
NO FUNCTION RAM	01	0	1	0	1	X	0	0	1	1	1	1
NO FUNCTION I/O	03	0	0	0	1	X	0	1	1	1	1	1

1* = positive pulse for 2 clk periods
LOW* = also reads M1 line on ERR
RUN* = also reads HALT line on ERR

Figure 7 - Control Function Codes

EZ80 Parts List

1	Z80 CMOS	3	1K 1/4 WATT RESISTOR
1	82C55 PPI	3	10K 1/4 WATT RESISTOR
1	HM62256 SRAM	10	.1 uF CERAMIC CAPACITOR
1	74HC02 QUAD 2 INPUT NOR	3	1 uF TANTALUM CAPACITOR
1	74HC14 HEX SCHMITT TRIGGER	1	100 uF 25V CAPACITOR
1	74HC139 DUAL 1 OF 4 DECODER	1	82 pF CERAMIC CAPACITOR
1	74HC257 4 X 2 MULTIPLEXER	1	220 pF CERAMIC CAPACITOR
1	74HC541 OCTAL BUFFER	1	4 MHz CRYSTAL
7	IC SOCKETS (OPTIONAL)	4	1N34A GERMANIUM DIODE
3	26 PIN MALE HEADER (.1)	1	RESET PUSH BUTTON
5	9 X 10K RESISTOR SIP	1	2N3904 TRANSISTOR
2	PC 26 PIN TO DB25 CABLE	1	64 KEY MATRIX KEYBOARD
1	EZ80 PCB	1	26 FEMALE IDC WITH CABLE
8	1N4448 SMALL SIGNAL DIODE	1	3 VOLT LITHIUM CELL

Don and Abed,

A couple things come to mind. It is possible that Word 97 can not read Wordstar files. Maybe you can try sending a text file. You create a text file in Pipedream by Saving as Plain Text. Text files on the Z88 and the PC are formatted differently. The ends of lines on the Z88 include a carriage return. The ends of lines on the PC include a carriage return and a line feed. PC-Link may convert the file correctly. If not you have to run the file through a conversion program. You say you can't find the file on the PC afterwards. It is possible that your files did not have the proper three letter extension. That is the only way PCs recognize files. Or you may have to look for the file with Find on the Start Menu.

Z88

I have a PC and a Mac. I do not have Word or PC-Link on the PC. I do have Mac-Link on the Mac. I can provide step by step directions to send files with PC-Link but I do not have PC-Link. If you want you can send me a copy of your PC-Link disk. I already have the Z88 PC-Link ROM (which is the same as the Mac-Link ROM) and a cable. I can borrow my laptop from work which has Word on it.

Here is a disk with Z88 communication files on it. You can use this instead of PC-Link to send files back and forth. You must install the Z88COM program first. Use HyperTerminal on your PC. It is built into Windows. I have included a shortcut to HyperTerminal. Z88.ht You must use the built-in Import-Export program on the Z88 and send the files phone.log and z88com.cli to the Z88. You send them as text files. Wait until the numbers have finished counting up. Once Z88com.cli is in your Z88 select it in Filer and choose Execute. You will see all the lines of the file scroll by one at a time. Once it has stopped at Line 5150 type on a new line SAVE "Z88com.bas" Now when you want to run it type RUN "z88com.bas" from BASIC. This procedure is also in the manual. Now try transferring some files with XModem. There is another XModem program on the disk zcp.bas. Try sending that over to the Z88. Select Transfer and Send File on the PC and Receive file on the Z88. Type the filenames on both machines and press enter at the same time on both machines. Now you will notice that it does not seem to be working. Don't be alarmed. Z88COM and ZCP use Checksum for Error Checking. HyperTerminal initially uses CRC but changes to Checksum on the third retry.

Actually if all you are doing is sending text files you only need the built-in HyperTerminal and Import-Export programs. But if you want to transfer BASIC programs then you need to use Z88COM or ZCP. You can convert the BASIC programs to text (cli) files but it is easier to transfer the BASIC programs as is. The disk also includes fadder.bas. This is a fast line feed adder for text files going from the Z88 to the PC. Kermit.bas is another transfer program which uses the kermit protocol. ZFU is a file utility to archive files. It is nice for backups. Finally I included the Patch to extend BASIC graphics commands. Let me know how you make out with the disk.

Dave Bennett

Unclassified Ads

Place your ads here, it is free!

Please inform and/or **update the Editor of any changes in your ad/s**

We have been a part of the Sinclair scene since 1982, repairing **ZX Spectrums** for Sinclair Research in England.
We provide Sales, Service, and Software for the

QL, Spectrum, ZX-81 and Z88

www.members.tripod.com/hes_computing/hes1.html
E-Mail 74601.1535@compuserve.com
Hours of Operation is Monday - Friday 1300 hrs. to 2100 hrs. central time zone.
Phone 210 661-4376

Home Electronics Service
John R. Rish
5222 Kazen Dr.
San Antonio TX 78219 USA

Pro Digital Electronics

323 SE 28th Ave.
Portland, Oregon 97214
503-232-3200

Pro Act Consulting, Inc.

2660 N. Houghton Rd.
Tucson, AZ 85749
520-749-5395, fax 520-749-3626
email <proactmd@aol.com>

The John Oliger Co.

11601 Widbey Dr.
Cumberland IN 46229

The John Oliger Floppy Disk System
FOR THE TS-2068
2068 User Cartridge
DISK BOARDS "A" & "B"
2068 Parallel Printer Port
2068/SPECTRUM Joystick Port
DFh Mapped Universal I/O Port board
User Manual only : $5.00 (Read before you buy)
joliger@mindspring.com

QLAMBer	$20	QLuMSi	$20
SeekQL	$10	Upgrades	$5

PLATYPUS SOFTWARE

Al Feng
914 Rio Vista Cir SW
Albuquerque NM 87105
(505) 843-8414

WOOD & WIND

ARCHIVE Based QL Software

QLerk - A complete financial program for the QL	
QLerk software (v3.21) with tutorial	$29
QLerk manual	$29
QLerk software & manual	$50
DBEasy - A menu based database system	
DBEasy software (v1.6)	$24
DBEasy upgrade from V1.5	$7
DBProgs - A toolkit of ARCHIVE procedures	
DBProgs software (v1.8)	$18
DBProgs upgrade from V1.7	$7
DBTutor - A general purpose learning program	
DBTutor software(v1.5)	$12
PC DBEasy - Just like QL DBEasy	
PC DBEasy software (v1.3)	$12

Bill Cable
Wood & Wind Computing
RR3 BOX 92
Cornish NH 03745 USA
Phone (603) 675-2218

Domino Cubes

Z88

Hardware & Software
352 7th Ave. 15th Flr.
New York, NY 10001
Phone 212 631-7563
Fax 212 947-5069 Voice mail pager 917 490-8407
Domino.cubes@excelsior.net

QL Hacker's Journal

Supporting All QL Programmers

Timothy Swenson, *Editor*
2455 Medallion Dr.
Union City, CA 94587-1914
swensontc@geocities.com
http://www.geocities.com/SilconValley/Pines/5865/

NESQLUG

NEWS

New England Sinclair QL Users Group

Ed Kingsley, Editor
16 Highland Avenue
Saugus MA 01906
(781) 233-3671 EdK4@aol.com

ZX-TEAM MAGAZIN

Peter Liebert-Adelt
LUETZOW STR 3
D-38102 BRAUNSCHWEIG
GERMANY
Email: p.liebert@t-online.de
http://home.t-online.de/home/p.liebert/zx-team.htm
Amateur Radio: DK4BF@DB0FC.#NDS.DEU.EU

The ZX Spectrum 48/128 Emulator

for IBM & Compatables: Z80 Version

Turn your PC into a real ZX Spectrum 48/128
⇒- Full Spectrum emulation, border, flash, beeper, Interface 1, Microdrive in cartridge file, RS232 input and output redirection to file, COM or LPT, joystick support, 128K sound through Soundblaster or internal speaker, built-in monitor,
⇒- Runs okay under DOS, Windows and DesqView,
Runs on any 640K PC; too slow for practical use on PC/XT's but fast enough on AT's, uses VGA/EGA/CGA or Hercules.
This program costs US $20. You will receive a 3.5" DD disk (5.25" disks on request), and you'll be kept informed about updates. Please send bank notes (bills), name and address to:

Gerton Lunter
PO Box 2535
NL-9704 CM Groningen
Netherland

Jochen Merz Software

SMSQ/E for the QXL

SMSQ/E for the Super GoldCard

QL Games & Upgrades QL Applications
ProWesS + Applications
Jochen Merz Software
Im stillen Winkel 12
47169 Duisburg, Germany
☎ 0203-502011 Fax 0203-502012
Credit Cards accepted
http://www.j-m-s.com/smsq/
e-mail smsq@j-m-s.com

Items for the Timex\Sinclair Computer

Timeworks Programming kit #1 For T/S 1000 & ZX81 $4.95
Mindware Gulp Game Timex 1000 & Sinclair ZX81 $4.95
Timex Horace & The Spiders for the 2068. $5.95
Chess (16K RAM) qty 5 price $2.95 ea
MC, VISA, American Express. Phone 717-748-1747

Keith Electronics

224 North Grove St
Lock Haven, PA. 17745

QL Today

QL Today is published by Jochen Merz Software. The representative in Britain is Miracle Systems Ltd. who take subscriptions and do the distribution.

English Office
Miracle Systems Ltd.
20 Mow Barton
Yates, Bristol, UK BS17 5NF
Tel. +44 1454 883602 Fax. +44 1454 883602

Editor
Dilwyn Jones
41 Bro Emrys
Tal-Y-Bont, Bangor, Gwynedd, UK LL57 3YT
Tel. +44 1248 354023 Fax. +44 1248 354023

Zebra Systems, Inc.

122 W26th St. Suite. 904
New York, NY 10001
Basics of Timex Sinclair 1500/1000 BASIC
BASIC Basics for the Timex/Sinclair 1500/1000
The Ins and Outs of the Timex TS-1000 & ZX-81
Computer Interfacing Technique in Science TS-1500/1000

Sinclair Resources

Keith Watson (AERCO & Z80 Emulator)
41634 Amberly Dr.
Mt. Clemens, MI 48038

Rod Gowen (RMG)
14784 S Quail Grove Cir
Oregon City OR 97045-8843

Surplus
T/S Inventory

JOHN J SHEPARD III
281 130th ST
OGDEN IA 50212
< jshepard@wccta.net >
Mostly QL & TS-2068

JACK BOATWRIGHT
67325 FRYREAR RD
BEND OR 97701
< jboatno4@outlawnet.com >
Mostly ZX-81/TS-1000 & TS-2068

QL TS-2068 ZX-81
Books Magazines
Software